SAMS

Освой самостоятельно

CSS

Расс Уикли

10 минут на урок

ВИЛЬЯМС

SAMS

Освой самостоятельно

CSS

Р.Уикли

10 минут на урок

SAMS
Освой самостоятельно

CSS

10 минут на урок

SAMS
Teach Yourself

CSS

Russ Weakley

in 10 Minutes

SAMS

800 East 96th Street, Indianapolis, Indiana, 46240 USA

SAMS
Освой
самостоятельно

CSS

Расс Уикли

10 минут
на урок

ВИЛЬЯМС
Издательский дом "Вильямс"
Москва • Санкт-Петербург • Киев
2006

ББК 32.973.26-018.2.75
У35
УДК 681.3.07

Издательский дом "Вильямс"

Зав. редакцией *С.Н. Тригуб*

Перевод с английского и редакция *А.В. Назаренко*

По сбщим вопросам обращайтесь в Издательский дом "Вильямс" по адресу: info@williamspublishing.com, http://www.williamspublishing.com
115419, Москва, а/я 783; 03150, Киев, а/я 152

Уикли, Расс.

У35 Освой самостоятельно CSS. 10 минут на урок. : Пер. с англ. — М. : Издательский дом "Вильямс", 2006. — 256 с. : ил. — Парал. тит. англ.

ISBN 5-8459-1019-6 (рус.)

Эта книга предназначена для тех пользователей, которым нужно быстро овладеть основами CSS — простого и мощного языка, применяемого для разработки стилей Web-страниц. Изучив предоставленную информацию, вы сможете сделать свои документы более надежными, компактными и доступными. Материал, представленный в виде простых задач и пошаговых инструкций, разбит на 22 урока, изучение каждого из которых отнимет не более 10 минут. Книга рассчитана как на начинающих, так и на опытных пользователей, которым необходимо быстро и эффективно освоить основные приемы работы с CSS. Если вы не собираетесь часами просиживать за объемными руководствами в поисках простейшей информации — эта книга для вас.

ББК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Sams Publishing.

Authorized translation from the English language edition published by Sams Publishing, Copyright © 2006.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the publisher.

Russian language edition is published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2006.

ISBN 5-8459-1019-6 (рус.)
ISBN 0-672-32745-7 (англ.)

© Издательский дом "Вильямс", 2006
© Sams Publishing, 2006

Оглавление

Введение 15
Урок 1. Что такое CSS 17
Урок 2. Использование правил CSS 21
Урок 3. Селекторы в действии 35
Урок 4. Применение стилей 43
Урок 5. Блочная модель CSS 51
Урок 6. Добавление фоновых изображений 59
Урок 7. Форматирование текста 67
Урок 8. Стилевое оформление гибкого заголовка 75
Урок 9. Заголовки в закругленной рамке 81
Урок 10. Стилевое оформление ссылок 87
Урок 11. Расположение изображения и его подписи 97
Урок 12. Создание фотогалереи 107
Урок 13. Стилевое оформление блока цитаты 117
Урок 14. Стилевое оформление таблицы 127
Урок 15. Создание вертикальной панели навигации 139
Урок 16. Создание горизонтальной панели навигации 149
Урок 17. Стилевое оформление рамки с закругленными углами 159
Урок 18. Создание заголовка сайта 173
Урок 19. Расположение двух колонок с заголовком и примечанием 185
Урок 20. Подготовка страницы для печати 205
Урок 21. Размещение трех колонок с заголовком и примечанием 217
Урок 22. Устранение ошибок CSS 241
Предметный указатель 252

Содержание

Введение 15

Для кого предназначена эта книга 15

Структура книги 15

Интерактивная поддержка 16

Договоренности, принятые в данной книге 16

Урок 1. Что такое CSS 17

Что такое CSS 17

Что означает “каскадные” 17

Зачем нужно использовать CSS 19

Резюме 20

Урок 2. Использование правил CSS 21

Подготовка HTML-кода 21

Создание набора правил 21

Использование нескольких объявлений 24

Объединение селекторов 26

Использование сокращенной формы записи свойств 27

Запись свойств границ в сокращенной форме 29

Задание полей и заполнения в сокращенной форме записи 31

Другие свойства в сокращенной форме записи 33

Резюме 34

Урок 3. Селекторы в действии 35

Подготовка HTML-кода 35

Селекторы типа 36

Селекторы класса 36

Селекторы идентификаторов 37

Селекторы-потомки 38

Универсальные селекторы 39
Расширенные селекторы 40
Резюме 42

Урок 4. Применение стилей 43
Подготовка HTML-кода 43
Применение внутритекстовых стилей 44
Использование стилей заголовков 45
Использование внешних таблиц стилей 46
Стили @import 47
Сокрытие стилей от старых браузеров 47
Резюме 49

Урок 5. Блочная модель CSS 51
Внутритекстовые и блочные элементы 51
Задание ширины блока 52
Поля 54
Цвет фона и фоновое изображение 55
Заполнение 55
Граница 56
Область содержимого 57
Резюме 58

Урок 6. Добавление фоновых изображений 59
Подготовка HTML-кода 59
Создание селекторов для определения стиля заголовка 60
Добавление фонового изображения 60
Настройка свойства background-repeat 61
Добавление свойства background-position 62
Использование сокращенной формы записи свойства background 64
Добавление заполнения 65
Резюме 66

Урок 7. Форматирование текста 67
Подготовка HTML-кода 67
Удаление элементов <font> 68
Создание селекторов 69
Формирование стиля элемента <p> 69
Стилевое оформление первого абзаца 71
Преобразование в сокращенную форму записи 72
Резюме 73

Урок 8. Стилевое оформление гибкого заголовка 75
Стилевое оформление заголовка 75
Добавление цвета, размера шрифта и ширины 76
Задание опций текста 78
Добавление заполнения и границ 78
Добавление фонового изображения 80
Резюме 80

Урок 9. Заголовки в закругленной рамке 81
Определение стиля заголовка 81
Стилевое оформление элемента <h2> 82
Добавление фонового изображения 83
Стилевое оформление элемента <em> 84
Резюме 85

Урок 10. Стилевое оформление ссылок 87
Ссылки и псевдоклассы 87
Установка порядка следования псевдоклассов 89
Использование классов с псевдоклассами 89
Стилевое оформление ссылок с фоновыми изображениями 90
Удаление подчеркивания и применение границ 92
Увеличение активной области ссылок 94
Резюме 95

Урок 11. Расположение изображения и его подписи 97
Оболочка для изображения и подписи 97
“Плавающий” контейнер 98
Применение заполнения, цвета фона и фонового изображения 100
Стилевое оформление подписи 102
Стилевое оформление изображения 103
Расположение подписи рядом с рисунком 104
Создание фотоснимка 105
Резюме 106
Урок 12. Создание фотогалереи 107
Создание галереи с эскизами 107
Расположение элементов <div> 108
Стилевое оформление изображения 110
Стилевое оформление элемента абзаца 111
Принудительный перевод строки 113
Размещение подписей рядом 115
Резюме 116
Урок 13. Стилевое оформление блока цитаты 117
Применение элемента <blockquote> 117
Стилевое оформление элемента <blockquote> 119
Стилевое оформление абзаца 121
Стилевое оформление класса source 122
Создание другого варианта 124
Резюме 126
Урок 14. Стилевое оформление таблицы 127
Стартовая таблица 127
Добавление к таблице элементов, повышающих ее доступность 128
Создание селекторов для формирования стилевого оформления таблицы 131

Стилевое оформление подписи 132
Стилевое оформление элемента <table> 133
Стилевое оформление элементов <th> и <td> 134
Стилевое оформление элемента <tr> 134
Выбор экземпляров элемента <th> 135
Чередование цветов в строках 137
Резюме 138

Урок 15. Создание вертикальной панели навигации 139

Зачем нужен список 139
Определение стиля списка 140
Стилевое оформление элемента <ul> 140
Стилевое оформление элемента <a> 141
Добавление эффекта визуального выделения 146
Стилевое оформление элемента <li> 147
Резюме 148

Урок 16. Создание горизонтальной панели навигации 149

Стилевое оформление списка 149
Стилевое оформление элемента <ul> 150
Стилевое оформление элемента <li> 152
Стилевое оформление элемента <a> 152
Определение стиля псевдокласса :hover 156
Резюме 157

Урок 17. Стилевое оформление рамки с закругленными углами 159

Подготовка HTML-кода 159
Создание иллюзии закругленных углов 160
Создание селекторов для определения стиля рамки с закругленными углами 161
Подготовка изображений 161
Стилевое оформление элемента <div> 162
Стилевое оформление элемента <h2> 163
Стилевое оформление элемента <p> 164

Стилевое оформление класса .furtherinfo 165
Стилевое оформление элемента <a> 166
Создание варианта с фиксированной длиной 168
Создание гибкого варианта только с верхним изображением 170
Резюме 172

Урок 18. Создание заголовка сайта 173
Подготовка HTML-кода 173
Создание селекторов для определения стиля заголовка 174
Стилевое оформление элемента <body> 175
Стилевое оформление контейнера 176
Стилевое оформление элемента <h1> 177
Стилевое оформление элемента <image> 178
Стилевое оформление элемента <ul> 179
Стилевое оформление элемента <li> 181
Стилевое оформление элемента <a> 183
Резюме 184

Урок 19. Расположение двух колонок с заголовком и примечанием 185
Подготовка HTML-кода 185
Создание селекторов для стилевого оформления двух колонок 186
Стилевое оформление элемента <body> 186
Стилевое оформление контейнера 187
Стилевое оформление элемента <h1> 189
Стилевое оформление контейнера #nav 191
Стилевое оформление элемента <ul> 192
Стилевое оформление элемента <li> 194
Стилевое оформление контейнера #content 196
Стилевое оформление контейнера #footer 197
Стилевое оформление элемента <h2> 200
Стилевое оформление элемента <a> 202
Резюме 204

Урок 20. Подготовка страницы для печати 205
Настройки CSS для печати 205
Начальные коды HTML и CSS 207
Создание селекторов для стиля для печати 208
Стилевое оформление элемента <body> 209
Стилевое оформление элемента <h1> 211
Стилевое оформление контейнера #nav 212
Стилевое оформление контейнера #footer 214
Стилевое оформление элемента <a> 215
Резюме 216

Урок 21. Размещение трех колонок с заголовком и примечанием 217
Подготовка HTML-кода 217
Создание селекторов для определения стиля страницы с тремя колонками 218
Создание сетки “жидкой” структуры 220
Создание фоновых изображений 220
Стилевое оформление элемента <body> 221
Стилевое оформление элемента <h1> 223
Стилевое оформление элементов <h2> и <h3> 225
Стилевое оформление первого контейнера 226
Стилевое оформление второго контейнера 228
Стилевое оформление колонки #content 229
Стилевое оформление колонки #news 231
Стилевое оформление колонки #nav 232
Стилевое оформление элемента <ul> 234
Стилевое оформление элемента #footer 236
Резюме 239

Урок 22. Устранение ошибок CSS 241
Подготовка кода CSS 241
Решение проблем 242
Советы по устранению проблем CSS 250
Резюме 252

Предметный указатель 253

Об авторе

Расс Уикли (Russ Weakley) работает в сфере дизайна более 18 лет. Последние девять лет он специализируется на Web-дизайне и имеет собственную фирму Max Design. Кроме того, он работает Web-дизайнером Австралийского музея.

Вместе с Питером Фермингером (Peter Firminger) Расс является сопредседателем Web Standards Group. Задача данной группы — помочь Web-дизайнерам в освоении новых технологий и вопросов, связанных с доступностью. Кроме того, Расс является соучредителем Web Essentials, организовывающей конференции и семинары Web-разработчиков, на которые привлекаются докладчики и делегаты со всего мира.

Расс известен благодаря своим презентациям и семинарам по Web-дизайну, стандартам и доступности. Кроме того, Уикли является автором популярных и востребованных руководств по CSS, включая Listamatic, Listamatic2, Listutorial, Floatutorial и Selectutorial, которые можно найти на его Web-сайте `http://www.maxdesign.com.au`.

Благодарности

Автор благодарит команду издательства Sams, в частности Линду Гаррисон (Linda Harrison), за предоставленную возможность написать книгу. Благодарю всех, кто комментировал мою работу, критиковал ее и давал полезные советы, касающиеся моих руководств. Эта обратная связь позволила мне глубже понять проблемы других дизайнеров и разработчиков, сталкивающихся с CSS. Спасибо Лизе Миллер (Lisa Miller), вычитавшей текст и протестировавшей весь приведенный в книге код. Наконец, благодарю моего партнера Анну (Anna) за терпение, поддержку и ободрение в процессе написания книги.

От издательства

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересны любые ваши замечания в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш Web-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Отправляя письмо или сообщение, не забудьте указать название книги и ее авторов, а также свой обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию новых книг.

Наши электронные адреса:

E-mail: info@williamspublishing.com
WWW: http://www.williamspublishing.com

Наши почтовые адреса:

в России: 115419, Москва, а/я 783
в Украине: 03150, Киев, а/я 152

Введение

Каскадные таблицы стилей (Cascading Style Sheets — CSS) — это простой и мощный язык определения стилей Web-страниц. Независимо от того, кем вы являетесь — Web-дизайнером, разработчиком или другим человеком, использующим Web, — CSS должны стать важной составляющей вашего набора инструментов для разработки Web-сайтов.

Многие разработчики Web-сайтов используют таблицы для формирования структуры и не понимают преимуществ CSS. Хотя существует множество хороших ресурсов и книг по CSS, люди часто теряются в огромном потоке информации. Часто бывает просто трудно понять, с чего же лучше начать обучение.

Книга "Освой самостоятельно CSS. 10 минут на урок" предназначена для того, чтобы помочь вам быстро разобраться в CSS с помощью ряда простых и последовательных уроков.

Для кого предназначена эта книга

Эта книга — для вас, если вы

- новичок в использовании CSS;
- нуждаетесь в простом руководстве по использованию CSS;
- желаете быстро узнать, как задействовать все возможности CSS;
- хотите узнать о новых способах применения CSS.

Структура книги

Книга "Освой самостоятельно CSS. 10 минут на урок" состоит из 22 уроков, в которых последовательно объясняются особенности CSS. Прочитав ее, вы будете хорошо понимать язык CSS и сможете использовать его в реальных ситуациях. Каждый урок состоит из простых этапов, благодаря которым вы быстро освоите новую концепцию и сможете применять ее на практике. Кроме того, уроки являются самодостаточными, поэтому при необходимости вы можете сразу переходить к интересующей вас теме.

Интерактивная поддержка

Для каждого урока книги “Освой самостоятельно CSS. 10 минут на урок” существуют файлы-примеры, которые можно загрузить с Web-сайта издательства (`http://www.williamspublishing.com/`).

Договоренности, принятые в данной книге

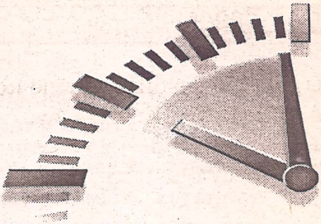
Чтобы отличать код HTML/CSS от другого содержимого, мы представили его `моноширинным` шрифтом. **Полужирным шрифтом** отмечены фрагменты, которыми данный код отличается от предшествующего.

Пиктограмма “Примечание” представляет важную информацию, связанную с текущей темой.

Пиктограмма “Предупреждение” сообщает о потенциальных проблемах, связанных с CSS или его реализациями в определенных браузерах.

Пиктограмма “Совет” отмечает подсказки или информацию, позволяющую немного облегчить вашу работу.

Урок 1



Что такое CSS

Из этого урока вы узнаете о каскадных таблицах стилей (Cascading Style Sheets — CSS) и о том, почему их необходимо использовать.

Что такое CSS

Каскадные таблицы стилей (Cascading Style Sheets — CSS) — это язык, используемый в документах HTML для определения способа представления содержимого. Представление задается с помощью стилей, помещаемых непосредственно в элементы HTML, заголовок HTML-документа или отдельные таблицы стилей.

Таблицы стилей содержат несколько правил CSS. Каждое правило выбирает элементы в HTML-документе. Затем данные правила определяют, каким стилем будут представлены элементы.

С одним файлом CSS может связывать любое число файлов HTML.

Что означает “каскадные”

Существует три типа таблиц стилей, которые могут повлиять на представление HTML-документа в браузере. Перечислим их.

- **Таблицы стилей браузера.** Браузеры применяют таблицы стилей ко всем Web-документам. Хотя эти таблицы стилей зависят от браузера, все они обладают общими характеристиками — черный текст, синие ссылки, коричневые посещенные ссылки. Данные таблицы стилей браузера являются таблицами *по умолчанию*.

- **Пользовательские таблицы стилей.** Пользователем является любой посетитель вашего Web-сайта. Большинство современных браузеров позволяет пользователям устанавливать свои таблицы стилей в браузерах. Данные стили имеют более высокий приоритет по сравнению с таблицами стилей браузера по умолчанию, но только для этого пользователя.
- **Авторские таблицы стилей.** Автором является тот, кто разработал Web-сайт, т.е. вы! Применив таблицу основного стиля к странице, вы добавили авторскую таблицу стилей. Обычно авторские таблицы стилей имеют более высокий приоритет, чем пользовательские (которые, в свою очередь, подавляют стили браузера). Данный каскад показан на рис. 1.1.

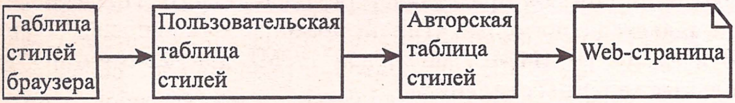


Рис. 1.1. *Три типа таблиц стилей, влияющих на представление Web-страницы*

Каскад означает, что стили могут ниспадать друг в друга (или образовывать каскад). Посредством каскада определяются приоритеты правил, которые позже применяются к документу. Например, правила авторских таблиц стилей обычно имеют более высокий приоритет, чем таблицы стилей браузера по умолчанию.

Откуда взялись CSS

CSS — это рекомендация Консорциума World Wide Web (World Wide Web Consortium — W3C). W3C — это промышленный консорциум сторон, заинтересованных в развитии WWW; в него входят университеты, такие компании, как Microsoft, Netscape и Macromedia, а также эксперты смежных областей.

Одной из задач W3C является выработка рекомендаций для различных аспектов деятельности Web. Технология CSS1 получила статус рекомендации в конце 1996 года, CSS2 стала рекомендацией в мае 1998 года, а CSS2.1 — в январе 2003 года.

Зачем нужно использовать CSS

Перечислим некоторые преимущества использования CSS для разработчиков Web-страниц.

- **Простата эксплуатации.** Истинная мощь CSS заключается в том, что один файл CSS может контролировать внешний вид множества документов HTML. Чтобы изменить вид всего сайта, достаточно отредактировать единственный файл CSS (а не огромный набор HTML-документов).
- **Меньшие размеры файлов.** CSS позволяет авторам удалять из HTML-документов все элементы, связанные с представлением, — таблицы стилей, разделительные изображения, декоративные изображения, шрифты, цвета, показатели ширины, высоты и фоновые изображения. Представления можно контролировать с помощью файлов CSS, что позволяет существенно уменьшить размеры файлов, представляющих HTML-документы.
- **Повышенная доступность.** Если объединить CSS с HTML-документами с правильной структурой, можно существенно облегчить работу таких средств, как программы чтения с экрана. Если удалить разметку, связанную с визуальным представлением, останется только структурное содержимое, что и требуется подобным программам. Кроме того, с помощью CSS можно увеличивать площадь ссылок, на которой можно щелкать, контролировать высоту и длину строки для пользователей с нарушенными моторными или когнитивными функциями.
- **Различные среды.** Стили CSS можно разрабатывать для различных сред — браузеров, принтеров, портативных

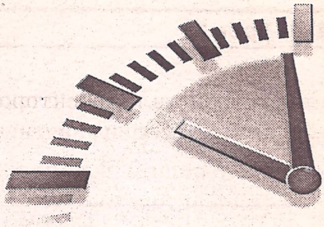
устройств и проекторов, не меняя при этом содержимое или структуру документа.

- **Повышенный контроль над типографией.** CSS позволяет авторам контролировать представление содержимого с помощью таких свойств, как `capitalize`, `uppercase`, `lowercase`, `text-decoration`, `letter-spacing`, `word-spacing`, `text-indent` и `line-height`. CSS также позволяет добавлять поля, границы, заполнение, фоновый цвет и фоновые изображения к любому элементу HTML.

Резюме

Из этого урока вы узнали о каскадных таблицах стилей (Cascading Style Sheets — CSS) и о том, зачем они нужны. Также вы узнали, откуда взялись таблицы стилей и какие три типа таблиц стилей могут влиять на внешний вид Web-страницы.

Урок 2



Использование правил CSS

В этом уроке вы освоите синтаксис и правила языка каскадных таблиц стилей (Cascading Style Sheet — CSS), выучите компоненты правил CSS, включая селекторы, объявления, свойства и значения, и научитесь сопоставлять стили с рядом простых элементов HTML. Кроме того, вы узнаете, как использовать сокращенную форму записи свойств.

Подготовка HTML-кода

HTML-код этого урока будет состоять из трех элементов — `<h1>`, `<h2>` и `<p>`, как показано в листинге 2.1.

Листинг 2.1. Код HTML, содержащий разметку для урока 2

```
<h1>
    Заголовок первого уровня
</h1>
<h2>
    Заголовок второго уровня
</h2>
<p>
    Текст абзаца ...
</p>
```

Создание набора правил

Правило (или *набор правил*) — это утверждение, сообщающее браузеру, как визуализировать определенные элементы HTML-страницы. Набор правил состоит из селектора, за которым следует блок объявления. Внутри блока объявления может быть одно или несколько объявлений, каждое из которых содержит свойство и значение, как показано на рис. 2.1.

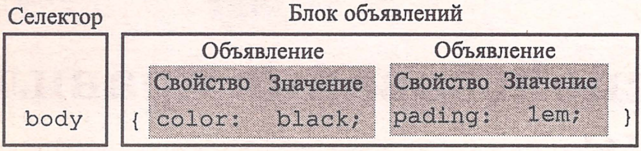


Рис. 2.1. Структура набора правил

При создании набора правил первым этапом является выбор селектора, отбирающего элементы HTML-страницы, на которые будет распространяться правило. Селектор состоит из всех символов до первой левой фигурной скобки (не включая ее). Селекторы, использованные в данном уроке, показаны в листинге 2.2. Более подробно селекторы обсуждаются в уроке 3, “Селекторы в действии”.

Листинг 2.2. Код CSS, иллюстрирующий селекторы

```
h1
h2
p
```

Далее необходимо создать блок объявления — контейнер, содержащий все данные между фигурными скобками (включая сами скобки). Блоки объявлений, использованные в этом уроке, представлены в листинге 2.3.

Листинг 2.3. Код CSS, иллюстрирующий блоки объявлений

```
h1 {...}
h2 {...}
p {...}
```

Внутри данного блока находится одно или несколько объявлений. Объявления сообщают браузеру, как нарисовать любой элемент на выбранной странице. Объявление состоит из свойства и одного или нескольких значений, разделенных двоеточиями. Конец каждого объявления обозначается точкой с запятой. Объявления, использованные в этом уроке, приведены в листинге 2.4.

Листинг 2.4. Код CSS, иллюстрирующий объявления

```
h1 { text-align: center; }
h2 { font-style: italic; }
p { color: maroon; }
```

Использование свободного места

Вокруг набора правил и внутри блоков объявлений допускается использование свободного места (пробелов, табуляций, символов перевода строки и возврата каретки).

Наборы правил можно выстраивать так, как вам удобно. Одни разработчики предпочитают давать все объявления в одной строке с целью экономии места, как показано в листинге 2.5. Другие размещают каждое объявление в отдельной строке, как показано в листинге 2.6, что делает их более удобочитаемыми.

Листинг 2.5. Код CSS, иллюстрирующий представление набора правил в одной строке

```
h2 { font-style: italic; text-align: center; color: navy; }
```

Листинг 2.6. Код CSS, иллюстрирующий представление набора правил в нескольких строках

```
h2
{
    font-style: italic;
    text-align: center;
    color: navy
}
```

Свойство — это один из аспектов элемента, который относится к стилю представления. В каждом объявлении может быть только одно свойство, если не используется сокращенная форма записи (см. раздел “Использование сокращенной формы записи свойств” далее в этом уроке). Свойства, используемые в этом уроке, выделены в листинге 2.7 полужирным шрифтом.

Листинг 2.7. Код CSS с примерами свойств

```
h1 { text-align: center; }
h2 { font-style: italic; }
p { color: maroon; }
```

Значение — это точный стиль, который вы желаете установить для свойства. Значениями могут быть `length`, `percentage`, `color`, `url`, `keyword` и `shape`. Значения, используемые в этом уроке, выделены в листинге 2.8 полужирным шрифтом.

Листинг 2.8. Код CSS, иллюстрирующий значения

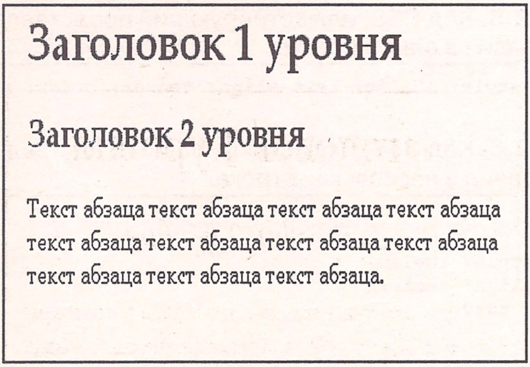
```
h1 { text-align: center; }
h2 { font-style: italic; }
p { color: maroon; }
```

Первый набор правил касается элемента `<h1>` и выравнивает его по центру окна браузера.

Второе правило касается элемента `<h2>` и представляет его курсивом.

Третье правило касается элемента `<p>` и окрашивает весь текст внутри этого элемента коричневым цветом.

Результаты применения стилей к HTML-коду, представленному в листинге 2.1, показаны на рис. 2.2.



Заголовок 1 уровня

Заголовок 2 уровня

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Рис. 2.2. Элементы с переопределенными стилями

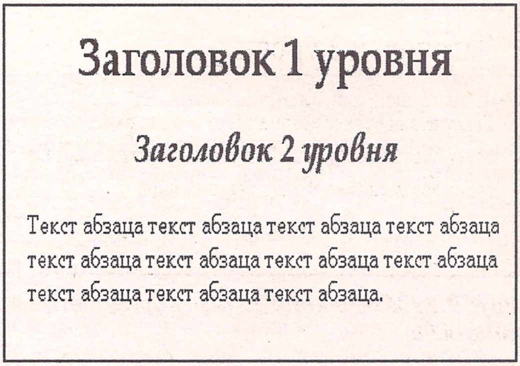
Использование нескольких объявлений

В блоке объявлений можно использовать более одного объявления. В таком случае объявления должны разделяться точкой с запятой.

В данном примере элементы `<h1>` и `<h2>` будут представлены согласно новому объявлению — `color: navy;`. Элемент `<h2>` также будет представлен согласно определению стиля `text-align: center;` (выравнивает текст по центру окна браузера). Новые объявления приведены в листинге 2.9. Результаты выполнения данного кода показаны на рис. 2.3.

Листинг 2.9. Код CSS, иллюстрирующий применение нескольких объявлений

```
h1
{
    text-align: center;
    color: navy;
}
h2
{
    font-style: italic;
    text-align: center;
    color: navy;
}
p
{
    color: maroon;
}
```



Заголовок 1 уровня

Заголовок 2 уровня

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Рис. 2.3. Переопределение стилей с несколькими объявлениями

Объединение селекторов

Если одинаковые объявления относятся к нескольким селекторам, их можно сгруппировать, чтобы не писать одно и то же правило несколько раз. Селекторы следует разделять запятыми. В нашем примере элементы <h1> и <h2> имеют два общих объявления, поэтому соответствующие части двух наборов правил можно объединить, как показано в листинге 2.10.

Листинг 2.10. Код CSS, иллюстрирующий объединение селекторов

```
h1, h2
{
    text-align: center;
    color: navy;
}
h2
{
    font-style: italic;
}
p
{
    color: maroon;
}
```

Добавление комментариев CSS

С целью объяснения кода к CSS можно добавлять комментарии. Подобно комментариям HTML, комментарии CSS будут игнорироваться браузером.

Комментарий CSS начинается символами `/*`, а заканчивается символами `*/`. Комментарии могут располагаться перед правилом, в правиле, а также занимать несколько строк. Кроме того, их можно использовать для пояснения всех правил или отдельных объявлений.

Использование сокращенной формы записи свойств

Сокращенная форма записи позволяет задавать в одном свойстве значения нескольких свойств, благодаря чему свойства становится легче записывать и использовать. Кроме того, с их помощью файлы CSS можно сделать более лаконичными.

Например, стиль элемента `<h2>` можно определить с помощью свойств `font-style`, `font-variant`, `font-weight`, `font-size`, `line-height` и `font-family`, как показано в листинге 2.11, или с помощью одного свойства `font`, как показано в листинге 2.12 и на рис. 2.4.

Большинство сокращенных свойств не требует расположения значения в строгом порядке. Тем не менее при использовании свойства `font` безопаснее задавать значения в порядке, определенном W3C: `font-style`, `font-variant`, `font-weight`, `font-size`, `line-height` и `font-family`.

При использовании в свойстве `font` значений `font-size` и `line-height` вначале следует задать `font-size`, затем указать левую косую черту (/) и задать `line-height`, как показано в листинге 2.12.

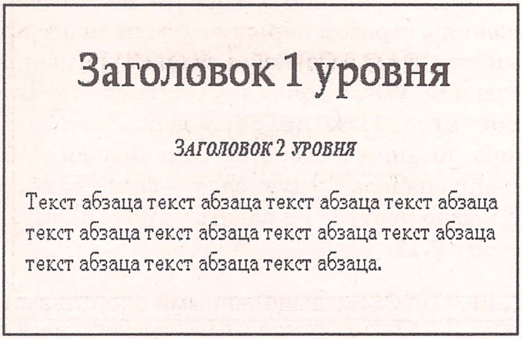
Листинг 2.11. Код CSS с выделенными свойствами шрифта

```
h1, h2
{
    text-align: center;
    color: navy;
}
h2
{
    font-style: italic;
    font-variant: small-caps;
    font-weight: bold;
    font-size: 100%;
    line-height: 120%;
    font-family: arial, helvetica, sans-serif;
}
p
{
    color: maroon;
}
```

Листинг 2.12. Код CSS с выделенными свойствами шрифта в сокращенной форме записи

```
h1, h2
{
    text-align: center;
    color: navy;
}
h2

{
    font: italic small-caps bold 100%/120% arial, helvetica,
sans-serif;
}
p
{
    color: maroon;
}
```



Заголовок 1 уровня

Заголовок 2 уровня

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

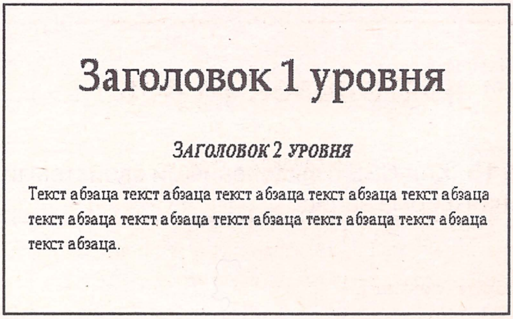
Рис. 2.4. Элемент <h2> с переопределенным стилем

Значениями свойств шрифта в сокращенной форме записи могут быть `font-style`, `fontvariant`, `font-weight`, `font-size`, `line-height` и `font-family`. Тем не менее в сокращенное объявление не обязательно включать их все. Например, для элемента `<p>` вы, возможно, желаете задать только значения `font-size` и `font-family`, как показано в листинге 2.13.

Листинг 2.13. Код CSS с выделенными свойствами шрифта

```
h1, h2
{
    text-align: center;
    color: navy;
}
h2
{
    font: italic small-caps bold 100%/120% arial, helvetica,
sans-serif;
}
p
{
    color: maroon;
    font: 80% arial, helvetica, sans-serif;
}
```

В этом случае `font-style`, `font-variant`, `font-weight` и `line-height` не включаются в сокращенное свойство и получают значения по умолчанию. Результат такого решения представлен на рис. 2.5.



Заголовок 1 уровня

ЗАГОЛОВОК 2 УРОВНЯ

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Рис. 2.5. *Элемент <p> с переопределенным стилем*

Запись свойств границ в сокращенной форме

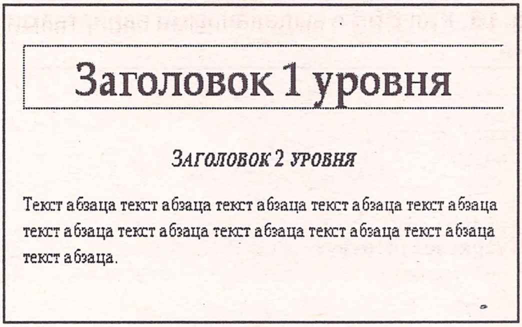
Свойства границ также можно представить с использованием сокращенной формы записи свойства `border`. Стиль элемента `<h1>` можно переопределить с помощью свойств `border-width`, `border-style` и `border-color`, как показано в листинге 2.14, или с помощью одного свойства `border`, как показано в листинге 2.15. Результаты такого решения представлены на рис. 2.6.

Листинг 2.14. Код CSS с выделенными свойствами границ

```
h1, h2
{
    text-align: center;
    color: navy;
}
h1
{
    border-width: 1px;
    border-style: solid;
    border-color: gray;
}
h2
{
    font: italic small-caps bold 100%/120% arial, helvetica,
sans-serif;
}
p
{
    color: maroon;
    font: 80% arial, helvetica, sans-serif;
}
```

Листинг 2.15. Код CSS с выделенными свойствами границ (сокращенная форма записи)

```
h1, h2
{
    text-align: center;
    color: navy;
}
h1
{
    border: 1px solid gray;
}
h2
{
    font: italic small-caps bold 100%/120% arial, helvetica,
sans-serif;
}
p
{
    color: maroon;
    font: 80% arial, helvetica, sans-serif;
}
```



Заголовок 1 уровня

Заголовок 2 уровня

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Рис. 2.6. Переопределенный стиль элемента <h1>

Задание полей и заполнения в сокращенной форме записи

Поля определяют расстояние между границами данного и соседних элементов. Заполнение создает пространство между границами элемента и его содержимым (подробнее об этом речь идет в уроке 5, “Блочная модель CSS”). Кроме того, свойства `margin` и `padding`, заданные в сокращенной форме, делают код CSS более компактным.

Свойство `margin` может объединять в себе `margin-top`, `margin-right`, `margin-bottom` и `margin-left`, а свойство `padding` — `padding-top`, `padding-right`, `padding-bottom` и `padding-left`.

Свойства `margin` и `padding` также можно использовать для определения различных стилей для всех сторон элемента. Значения применяются в следующем порядке: вверху (`top`), справа (`right`), внизу (`bottom`) и слева (`left`) — по часовой стрелке, начиная с верхнего положения.

Стиль элемента `<p>` можно изменить с помощью свойст в `padding-top`, `padding-right`, `padding-bottom` и `padding-left`, как показано в листинге 2.16, или с помощью одного свойства `padding`, как показано в листинге 2.17.

Листинг 2.16. Код CSS с выделенными свойствами заполнения

```
h1, h2
{
    text-align: center;
    color: navy;
}
h1
{
    border: 1px solid gray;
}
h2
{
    font: italic small-caps bold 100%/120% arial, helvetica,
sans-serif;
}
p
{
    color: maroon;
    font: 80% arial, helvetica, sans-serif;
    padding-top: 1em;
    padding-right: 2em;
    padding-bottom: 3em;
    padding-left: 4em;
}
```

Листинг 2.17. Код CSS с выделенными свойствами заполнения (сокращенная форма записи)

```
h1, h2
{
    text-align: center;
    color: navy;
}
h1
{
    border: 1px solid gray;
}
h2
{
    font: italic small-caps bold 100%/120% arial, helvetica,
sans-serif;
}
p
{
    color: maroon;
    font: 80% arial, helvetica, sans-serif;
    padding: 1em 2em 3em 4em;
}
```

В сокращенном объявлении вы можете использовать от одного до трех значений.

Объявление `p { padding: 1em; }` равнозначно применению заполнения `1em` ко всем сторонам элемента.

Объявление `p { padding: 1em 2em; }` равнозначно применению заполнения `1em` к верхней и нижней сторонам элемента и `2em` — к левой и правой сторонам.

Объявление `p { padding: 1em 2em 3em; }` равнозначно применению заполнения `1em` к верхней стороне, заполнения `2em` — к левой и правой сторонам и заполнения `3em` — к нижней стороне элемента.

Объявление `p { padding: 1em 2em 3em 4em; }` равнозначно применению заполнения `1em` к верхней стороне, `2em` — к правой стороне, `3em` — к нижней стороне и `4em` — к левой стороне элемента.

Другие свойства в сокращенной форме записи

Свойство `background` объединяет в себе `background-color`, `background-image`, `background-repeat`, `background-attachment` и `background-position`, как показано в листинге 2.18.

Листинг 2.18. Код CSS с выделенным свойством фона (сокращенная форма записи)

```
h1, h2
{
    text-align: center;
    color: navy;
}
h1
{
    border: 1px solid gray;
    background: yellow url(tint.jpg) repeat-y 100% 0;
}
h2
{
    font: italic small-caps bold 100%/120% arial, helvetica,
sans-serif;
}
```

```
p
{
    color: maroon;
    font: 80% arial, helvetica, sans-serif;
    padding: 1em 2em 3em 4em;
}
```

Свойство `list-style` объединяет `list-style-type`, `list-styleposition` и `list-style-image`, как показано в листинге 2.19.

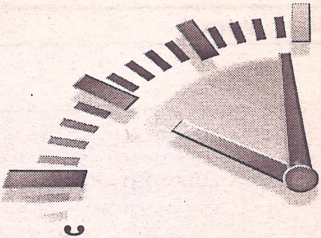
Листинг 2.19. Код CSS с выделенными свойствами списка (сокращенная форма записи)

```
h1, h2
{
    text-align: center;
    color: navy;
}
h1
{
    border: 1px solid gray;
    background: yellow url(tint.jpg) repeat-y 100% 0;
}
h2
{
    font: italic small-caps bold 100%/120% arial, helvetica,
sans-serif;
}
p
{
    color: maroon;
    font: 80% arial, helvetica, sans-serif;
    padding: 1em 2em 3em 4em;
}
ul
{
    list-style: square inside;
}
```

Резюме

Из этого урока вы узнали, как использовать селекторы, объявления, свойства, сокращенную форму записи свойств и значения для определения стилей ряда простых элементов HTML. Из следующего урока вы узнаете о типах селекторов и научитесь их использовать.

Урок 3



Селекторы в действии

Из этого урока вы узнаете о различных типах селекторов и научитесь их использовать.

Подготовка HTML-кода

Селекторы представляют собой один из важнейших аспектов CSS, поскольку используются для выбора элементов HTML-страницы и определения для них различных стилей. HTML-код этого урока показан в листинге 3.1.

Листинг 3.1. Код HTML, содержащий разметку для урока 3

```
<body>
<div id="content">
    <h1>
        Heading here
    </h1>
    <p class="intro">
        Lorem ipsum dolor sit amet.
    </p>
    <p>
        Lorem ipsum dolor <a href="#">sit</a> amet.
    </p>
</div>
<div id="nav">
    <ul>
        <li><a href="#" class="intro">item 1</a></li>
        <li><a href="#">item 2</a></li>
        <li><a href="#">item 3</a></li>
    </ul>
</div>
<div id="footer">
    Lorem ipsum dolor <a href="#">sit</a> amet.
</div>
</body>
```

Что такое `<div>`

Элемент `<div>` — это общий контейнер, который можно использовать для добавления структуры к документу HTML. Хотя он является элементом блочного уровня, никаким иным образом с представлением документа он не связан.

В данном примере элемент `<div>` использовался для создания таких логических единиц контекста, как панель навигации и примечание.

Далее можно определять стили для названных единиц содержимого, используя селекторы-потомки, рассмотренные ниже в этом уроке.

Селекторы типа

Селекторы типа выбирают любой элемент HTML на странице, согласующийся с селектором.

В примере HTML-кода, приведенном в листинге 3.1, содержится семь элементов HTML, которые можно использовать, как селекторы типа: `<body>`, `<div>`, `<h1>`, `<p>`, `<ul>`, `<li>` и `<a>`.

Например, чтобы выбрать все элементы `<li>` на странице, используется селектор `<li>`, как показано в листинге 3.2.

Листинг 3.2. Код CSS, содержащий стилевое оформление элемента `<li>`

```
li
{
    color: blue;
}
```

Селекторы класса

Селекторы класса можно использовать для выбора любого элемента, которому был присвоен атрибут класса.

В примере HTML-кода, приведенном в листинге 3.1, имеется два элемента HTML с атрибутами класса — `<p class="intro">`

и `<a href="#" class="intro">`. Чтобы, например, выбрать все экземпляры класса `intro`, применяется селектор `.intro`, как показано в листинге 3.3.

Листинг 3.3. Код CSS, содержащий стилевое оформление класса `.intro`

```
.intro
{
    font-weight: bold;
}
```

Кроме того, объединяя селекторы типа и класса, можно выбирать конкретный экземпляр класса. Например, вы можете пожелать по отдельности выбрать `<p class="intro">` и `<a href="#" class="intro">`. Для этого, как показано в листинге 3.4, можно использовать `p.intro` и `a.intro`.

Листинг 3.4. Код CSS, содержащий два различных стилевых оформления класса `.intro`

```
p.intro
{
    color: red;
}
a.intro
{
    color: green;
}
```

Селекторы идентификаторов

Селекторы идентификаторов похожи на селекторы классов. Их можно использовать для выбора любого элемента HTML, имеющего идентифицирующий атрибут. Тем не менее каждый идентификатор можно использовать в документе единственный раз, тогда как классы можно применять сколь угодно часто.

В этом уроке идентификаторы используются для определения уникальных частей структуры документа — оглавления, панели навигации и примечания.

В примере HTML-кода, приведенном в листинге 3.1, имеется три идентификатора: `<div id="content">`, `<div id="nav">`

и `<div id="footer">`. Чтобы выбрать `<div id="nav">`, используется селектор `#nav`, как показано в листинге 3.5.

Листинг 3.5. Код CSS, содержащий селектор `#nav`

```
#nav
{
    color: blue;
}
```

Что использовать: идентификатор или класс

Классы можно использовать в документе многократно. Любой идентификатор можно применять в документе только один раз. Если вам требуется несколько раз использовать один и тот же селектор, выбирайте классы.

Тем не менее идентификаторы имеют больший вес, чем классы. Если свойство для одного и того же элемента определяет и селектор класса, и селектор идентификатора, будет использовано значение селектора идентификатора. Например, `h2#intro { color: red; }` заменит собой `h2.intro { color: blue; }`.

Селекторы-потомки

Селекторы-потомки используются для выбора элементов, являющихся потомками другого элемента. Например, в примере HTML-кода, показанного в листинге 3.1, три элемента `<a>` являются потомками элементов `<li>`. Чтобы выбрать только эти три элемента `<a>`, не затронув все остальные, можно использовать селектор-потомок, как показано в листинге 3.6. Данный селектор нацелен только на элемент `<a>`, вложенный в элемент `<li>`.

Листинг 3.6. Код CSS, содержащий селектор-потомок

```
li a
{
    color: green;
}
```

Селекторы-потомки не обязаны использовать элементы, являющиеся прямыми потомками. Например, элемент `<a>` является потомком `<div id="nav">`, а также потомком элемента `<li>`. Это означает, что в качестве селектора можно использовать и `#nav` (листинг 3.7).

Листинг 3.7. Код CSS, содержащий селектор-потомок

```
#nav a
{
   color: red;
}
```

Селекторы-потомки также могут включать несколько уровней потомков, что, как показано в листинге 3.8, позволяет создавать более избирательные селекторы.

Листинг 3.8. Код CSS, содержащий селектор-потомок

```
#nav ul li a
{
   color: green;
}
```

Универсальные селекторы

Универсальные селекторы используются для выбора любого элемента. Например, чтобы установить поля и заполнение всех элементов равными 0, можно, как показано в листинге 3.9, использовать селектор `*`.

Листинг 3.9. Код CSS, содержащий универсальный селектор

```
*
{
   margin: 0;
   padding: 0;
}
```

Кроме того, универсальные селекторы можно использовать для выбора всех компонентов внутри другого элемента, как показано в листинге 3.10. В результате выполнения этого кода будут выбраны элементы, находящиеся внутри элемента `<p>`.

Листинг 3.10. Код CSS, содержащий универсальный селектор выбора всех элементов внутри `<p>`

```
p *
{
    color: red;
}
```

Расширенные селекторы

Порожденные селекторы (child selector) используются для выбора элемента, являющегося прямым потомком другого элемента (родителя). Порожденные селекторы выбирают не все элементы-потомки, а только прямые порождения. Например, вам требуется отобрать элемент `<em>`, являющийся прямым потомком элемента `<div>`, но не другие элементы `<em>`, являющиеся потомками `<div>`. Соответствующий селектор показан в листинге 3.11.

Порожденные селекторы не поддерживаются Windows Internet Explorer 5, 5.5 и 6, но поддерживаются большинством других стандартных браузеров.

Листинг 3.11. Код CSS, содержащий селектор прямого потомка

```
div > em
{
    color: blue;
}
```

Селекторы соседних братьев выбирают элементы-братья, следующие непосредственно за элементом. Например, требуется отобрать элементы `<h3>`, непосредственно следующие за элементами `<h2>`. Данная задача возникает довольно часто в реальном мире, поскольку между размещенными рядом элементами `<h2>` и `<h3>` (заголовками второго и третьего уровней) получается довольно большой отступ. Соответствующий селектор показан в листинге 3.12.

Селекторы соседних братьев не поддерживаются в Windows Internet Explorer 5, 5.5 и 6, но поддерживаются большинством других стандартных браузеров.

Листинг 3.12. Код CSS, содержащий селектор соседнего брата

```
h2 + h3
{
   margin: -1em;
}
```

Селекторы атрибутов используются для выбора элементов на основе их атрибутов или значений атрибутов. Например, может потребоваться выбрать все изображения HTML-страницы, именуемые `"small.gif"`, как показано в листинге 3.13.

Селекторы атрибутов не поддерживаются Windows Internet Explorer 5, 5.5 и 6, а также Macintosh Internet Explorer 5. Кроме того, они не поддерживаются ранними версиями Opera.

Листинг 3.13. Код CSS, содержащий селектор атрибутов

```
img[src="small.gif"]
{
   border: 1px solid #000;
}
```

Псевдоэлементы позволяют определять стиль информации, недоступной в дереве документа. Например, используя стандартные селекторы, никак нельзя определить стиль первой буквы или первой строки содержимого элемента. Тем не менее, как показано в листинге 3.14, это можно сделать с помощью псевдоэлементов.

Псевдоэлементы `:before` и `:after` не поддерживаются Windows Internet Explorer 5, 5.5 и 6 и Macintosh Internet Explorer 5. Кроме того, они не поддерживаются ранними версиями Opera.

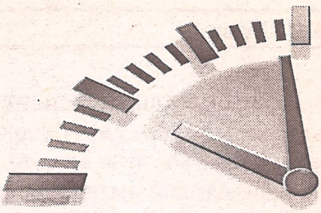
Листинг 3.14. Код CSS, содержащий селектор псевдоэлементов

```
p:first-line
{
   font-weight: bold;
}
p:first-letter
{
   font-size: 200%; font-weight: bold;}
```

Псевдоклассы `:first-child`, `:link`, `:visited`, `:hover`, `:active`, `:focus` и `:lang(n)` также позволяют форматировать элементы, отсутствующие в дереве документа. Подробнее псевдоклассы рассмотрены в уроке 10, “Стилевое оформление ссылок”.

Резюме

Из этого урока вы узнали, как использовать множество селекторов, в том числе селекторы типа, класса, идентификатора, потомка и универсальный селектор. Из следующего урока вы узнаете, как форматировать внутритекстовые элементы, определять стили заголовков и стили внутри внешних таблиц стилей.



Урок 4

Применение стилей

Из этого урока вы узнаете, куда можно поместить код CSS, включая текст, заголовок и внешние таблицы стилей. Вы также узнаете, как с помощью типа среды подготовить таблицу стилей для конкретного устройства (мобильного телефона, телевизора или КПК). Кроме того, вы узнаете о методах, позволяющих с помощью типа среды “спрятать” от старых браузеров новые методы задания стилей.

Подготовка HTML-кода

HTML-код этого урока, содержащий один абзац текста, показан в листинге 4.1.

Листинг 4.1. Код HTML, содержащий разметку для урока 4

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
        "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html;
    charset=utf-8">
    <title>Lesson 4</title></head>
<body>
<p>
    Lorem ipsum dolor sit amet, consectetuer adipiscing
    elit...
</p>
</body>
</html>
```

В этом уроке мы определим стиль элемента `<p>`, используя свойства `font-family`, `width`, `background-color`, `margin` и `padding`.

Данные стили можно применять к элементам <p>, используя три метода: внутритекстовые стили, стили заголовков и внешние таблицы стилей. Хотя описаны будут все методы, предпочтительным является использование внешних таблиц стилей, поскольку они не добавляют каскадные таблицы стилей к разметке HTML.

Применение внутритекстовых стилей

Внутритекстовые стили можно применять непосредственно к элементам HTML-кода, используя атрибут `style`. Тем не менее, если это возможно, внутритекстовых стилей следует избегать, поскольку в этом случае стили добавляются к разметке HTML. Это противоречит основной цели CSS: применять одинаковые стили к как можно большему числу страниц Web-сайта с помощью внешних таблиц стилей. Наличие стилей, которые применяются внутри текста, может стать источником дополнительной работы при модификации Web-сайта, поскольку придется изменять не один файл CSS, а множество страниц. Пример внутритекстового стиля показан в листинге 4.2.

Листинг 4.2. Код HTML, содержащий внутритекстовый стиль для элемента <p>

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
        "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html;
    charset=utf-8">
    <title>Lesson 4</title>
</head>
<body>
<p style="font-family: arial, helvetica, sans-serif;
argin:
1em;padding: 1em; background-color: gray; width: 10em;">
    Lorem ipsum dolor sit amet, consectetuer adipiscing
    elit...
</p>
</body>
</html>
```

Использование стилей заголовков

Для определения стиля элемента <p> могут также использоваться стили заголовков. С помощью элемента `style` в заголовок документа можно помещать правила CSS. Как и внутритекстовых стилей, стилей заголовка следует избегать, поскольку в этом случае стили добавляются в разметку HTML, а не во внешний файл CSS.

Иногда стили заголовка использовать выгодно, например когда правило CSS ограничивается одной страницей большого Web-сайта. Вместо того чтобы добавлять это правило ко всему файлу CSS, можно использовать стиль заголовка.

Пример стиля заголовка показан в листинге 4.3. Чтобы браузеры распознавали тип файла, в элементе `style` необходимо задать `type= "text/css"`.

Листинг 4.3. Код HTML, содержащий стили заголовка

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
        "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html;
    charset=utf-8">
    <title>Lesson 4 - listing 2</title>
<style type="text/css" media="screen">
    p
    {
        font-family: arial, helvetica, sans-serif;
        margin: 1em;
        padding: 1em;
        background-color: gray;
        width: 10em;
    }
</style>
</head>
<body>
<p>
    Lorem ipsum dolor sit amet, consectetuer adipiscing
    elit...
</p>
</body>
</html>
```

Использование внешних таблиц стилей

Третий метод определения стилей для документа включает связывание с внешними таблицами стилей. Данный метод формирования стилевого оформления является наиболее подходящим. Если стили потребуется изменить, модификации подвергается один файл CSS, а не все HTML-страницы.

Чтобы заменить стиль заголовка внешним стилем, набор правил необходимо переместить в новый файл CSS, как показано в листинге 4.4.

После этого из вашего HTML-файла необходимо с помощью элемента `link` связаться с требуемой таблицей стилей, как показано в листинге 4.5.

Листинг 4.4. Код CSS, содержащий внешнюю таблицу стилей со стилями для элемента <p>

```
p
{
    font-family: arial, helvetica, sans-serif;
    margin: 1em;
    padding: 1em;
    background-color: gray;
    width: 10em;
}
```

Листинг 4.5. Код HTML, содержащий элемент `link`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
        "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
    <meta http-equiv="content-type" content="text/html;
    charset=utf-8">
    <title>Lesson 4</title>
    <link rel="stylesheet" href="style.css" type="text/css"
    media="screen">
</head>
<body>
<p>
    Lorem ipsum dolor sit amet, consectetuer adipiscing
    elit...
</p>
</body>
</html>
```

Стили @import

Стили заголовков и внешние таблицы стилей также могут импортировать другие таблицы стилей, используя правило `@import`, как показано в листинге 4.6. Правило `@import` должно размещаться перед всеми остальными правилами заголовка или внешней таблицы стилей.

Листинг 4.6. Код CSS, содержащий импортированную таблицу стилей

```
@import "advanced.css";
p
{
    font-family: arial, helvetica, sans-serif;
    margin: 1em;
    padding: 1em;
    background-color: gray;
    width: 10em;
}
```

С помощью импортированных стилей можно использовать несколько файлов CSS, а также скрывать стили из старых браузеров.

Сокрытие стилей от старых браузеров

Некоторые старые браузеры, например Netscape Navigator 4 и Internet Explorer 4, обладают плохой поддержкой CSS. С помощью специальных типов среды и правил `@import` можно скрыть стили от этих браузеров.

Чтобы спрятать все стили от Netscape Navigator 4, необходимо заменить тип среды `screen` типом `screen, projection`, как показано в листинге 4.7. Поскольку Netscape Navigator 4 не поддерживает несколько типов среды, проблема будет решена.

Листинг 4.7. Код HTML, содержащий элемент `link` с типом среды `screen, projection`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
        "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
```

```
        <meta http-equiv="content-type" content="text/html;
        charset=utf-8">
        <title>Lesson 4</title>
    <link rel="stylesheet" href="style.css" type="text/css"
    media="screen, projection">
</head>
<body>
<p>
        Lorem ipsum dolor sit amet, consectetuer adipiscing
        elit...
</p>
</body>
</html>
```

Чтобы скрыть остальные стили от Internet Explorer 4 и нескольких других старых браузеров, набор правил для элемента <p> необходимо вынести из исходной таблицы стилей в импортированную, как показано в листингах 4.8 и 4.9. Поскольку Internet Explorer 4 не умеет читать импортированные файлы, проблема будет решена.

Листинг 4.8. Код CSS внутри исходной таблицы стилей вызывает `style.css`

```
@import "advanced.css";
```

Листинг 4.9. Код CSS внутри импортированной таблицы стилей вызывает `advanced.css`

```
p
{
    font-family: arial, helvetica, sans-serif;
    margin: 1em;
    padding: 1em;
    background-color: gray;
    width: 10em;
}
```

Все современные браузеры могут читать несколько типов среды (`screen`, `projection`), а также знают, как поступать с импортированными стилями, поэтому они отображают элемент <p> с полным стилевым оформлением.

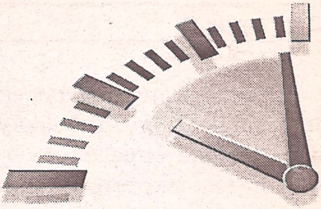
Чтобы скрыть стили заголовков от старых браузеров, можно заключить содержимое элемента `style` в комментарий, как показано в листинге 4.10.

Листинг 4.10. Код HTML, содержащий стили заголовка внутри комментария

```
<style type="text/css" media="screen">
<!--
    p
    {
        font-family: arial, helvetica, sans-serif;
        margin: 1em;
        padding: 1em;
        background-color: gray;
        width: 10em;
    }
-->
</style>
```

Резюме

В этом уроке вы научились применять к документу внутритекстовые стили, стили заголовка и внешние таблицы стилей. Кроме того, вы узнали, что такое тип среды и как скрыть сложные стили от старых браузеров с помощью нескольких типов среды и правила `@import`. В следующем уроке вы узнаете о блочной модели CSS, включающей поля, цвет фона, фоновое изображение, заполнение и границу.



Урок 5

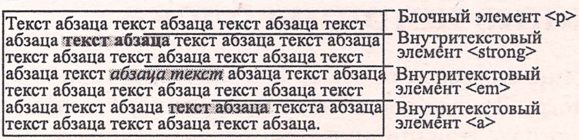
Блочная модель CSS

Из этого урока вы узнаете о блочной модели CSS, характеризуемой прямоугольниками, которые генерируются для всех элементов HTML, об аспектах, которые составляют модель рамки, включая поля, цвет фона, фоновое изображение, заполнение и границу, а также чем отличаются внутритекстовые элементы и элементы блочного уровня.

Внутритекстовые и блочные элементы

Элементы блочного уровня обычно отображаются в виде блоков с разрывом строки до и после элемента. В качестве примеров элементов блочного уровня можно привести абзацы, заголовки, элементы `div` и блочные цитаты.

Внутритекстовые элементы не отображаются, как блоки. Содержимое такого типа представляется в строках без разрыва строки до и после элемента. В качестве внутритекстовых элементов можно привести текст, выделенный курсивом или полужирным начертанием, и ссылки. Примеры блочных и внутритекстовых элементов показаны на рис. 5.1.


Заголовок
Блочный элемент <h1>
Блочный элемент <p>
Внутритекстовый элемент <strong>
Внутритекстовый элемент <em>
Внутритекстовый элемент <a>

Текст абзаца текст абзаца текст абзаца текст абзаца **текст абзаца** текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст *абзаца текст* абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текста абзаца текст абзаца текст абзаца текст абзаца.

Рис. 5.1. *Примеры блочных и внутритекстовых элементов*

Все блочные и внутритекстовые элементы являются блоками, использующими блочную модель. Стили элементов обоих типов можно определять с помощью таких свойств блочной модели, как `margin`, `background-color`, `background-image`, `padding` и `border`, как показано на рис. 5.2.

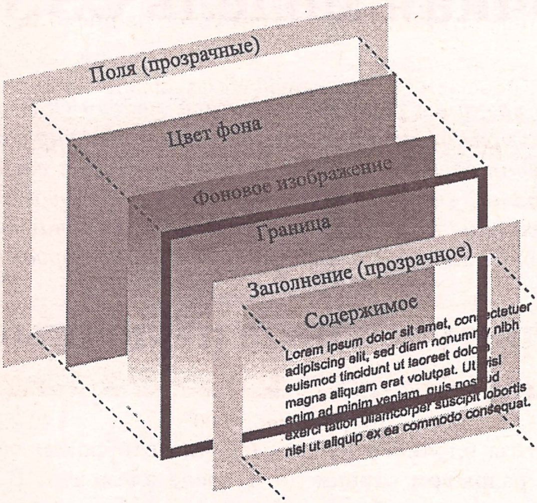

Поля (прозрачные)
Цвет фона
Фоновое изображение
Граница
Заполнение (прозрачное)
Содержимое
Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh
euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat. Ut wisi
enim ad minim veniam, quis nostrud
exerci tation ullamcorper suscipit lobortis
nisl ut aliquip ex ea commodo consequat.

Рис. 5.2. Трехмерная схема блочной модели CSS

Некоторые свойства блочной модели, например `height` и `width`, не применяются к внутритекстовым элементам. Кроме того, применение свойств `margin` и `padding` к внутритекстовым элементам влияет на содержимое, расположенное слева и справа от элемента, но не влияет на содержимое сверху и снизу.

Задание ширины блока

Ширина элемента определяет область, доступную для содержимого. Другие элементы (заполнение, граница и поля) добавляются к этой ширине.

Например, если элемент задан со свойством `width: 200px;`, ширина области содержимого равна 200 пикселям.

Если к тому же элементу применяется заполнение, граница или поля, они добавляются к общей ширине.

Однако Internet Explorer 5 для Windows (а в некоторых случаях и Internet Explorer 6 для Windows) использует иной метод задания ширины рамок. Если к элементу применяются заполнение и граница, их размеры вычитаются из общей ширины. Данная ситуация иллюстрируется на рис. 5.3.

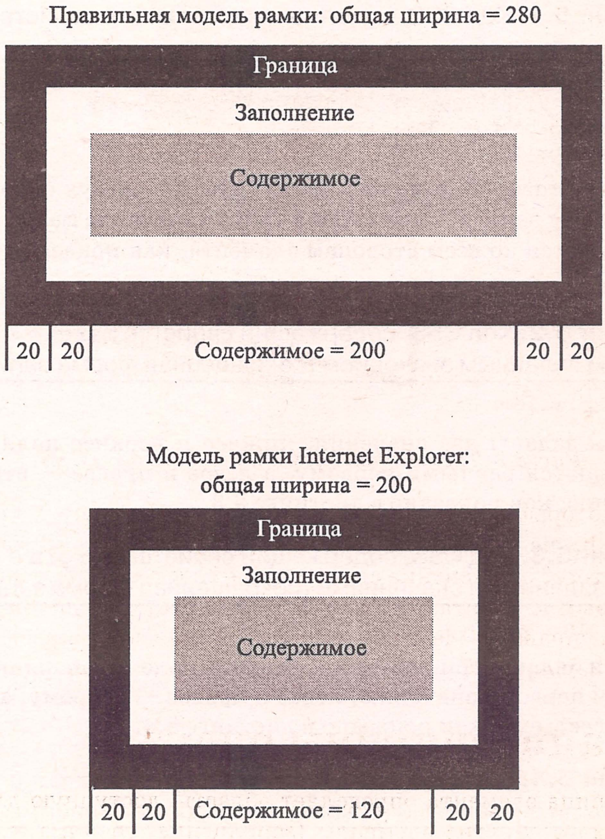

Правильная модель рамки: общая ширина = 280
Граница
Заполнение
Содержимое
20
20
Содержимое = 200
20
20
Модель рамки Internet Explorer:
общая ширина = 200
Граница
Заполнение
Содержимое
20
20
Содержимое = 120
20
20

Рис. 5.3. Блочная модель CSS и проблема ширины Internet Explorer

Поля

Поля можно применять к внешней части любого блочного или внутритекстового элемента. Они создают пустое пространство между краями данного и соседнего элементов.

Как показано в листинге 5.1, поля можно применять к отдельным сторонам блока.

Листинг 5.1. Код CSS, содержащий различные свойства `margin`

```
p { margin-top: 0; }
p { margin-right: 2em; }
h2 { margin-bottom: 3em; }
h3 { margin-left: 1em; }
```

Кроме того, поля можно применять, используя сокращенную форму записи. Если задано только значение `margin`, оно применяется ко всем сторонам элемента, как показано в листинге 5.2.

Листинг 5.2. Код CSS, содержащий свойство `margin` с одним заданным значением (сокращенная форма записи)

```
p { margin: 1em; }
```

Если заданы два значения, нижнее и верхнее поля устанавливаются равными первому, а левое и правое — второму значению, как показано в листинге 5.3.

Листинг 5.3. Код CSS, содержащий свойство `margin` с двумя заданными значениями (сокращенная форма записи)

```
p { margin: 1em 0; }
```

Если заданы три значения, верхнее поле устанавливается равным первому значению, левое и правое — второму, а нижнее — третьему, как показано в листинге 5.4.

Листинг 5.4. Код CSS, содержащий свойство `margin` с тремя заданными значениями (сокращенная форма записи)

```
p { margin: 1em 0 2em; }
```

Если заданы четыре значения, они применяются к верхнему, правому, нижнему и левому полям, как показано в листинге 5.5.

Листинг 5.5. Код CSS, содержащий свойство `margin` с четырьмя заданными значениями (сокращенная форма записи)

```
p { margin: 1em 2em 2em 1em; }
```

Цвет фона и фоновое изображение

Свойство `background-color` устанавливает цвет фона элемента.

Свойство `background-image` применяет к элементу фоновое изображение, которое размещается поверх фона, заданного с помощью `background-color`. Подробнее фоновые изображения рассмотрены в уроке 6, "Добавление фоновых изображений".

Заполнение

Заполнение можно применять к внешним границам области содержимого любого блочного или внутритекстового элемента. Заполнение создает пустое пространство между границей элемента и его содержимым.

Подобно полям, заполнение можно применять к отдельным сторонам блока, как показано в листинге 5.6.

Листинг 5.6. Код CSS, содержащий различные свойства заполнения

```
p { padding: 1em; }
h1 { padding-top: 0; }
h2 { padding-right: 2em; }
h2 { padding-bottom: 3em; }
h3 { padding-left: 1em; }
```

Кроме того, заполнение можно применять с использованием сокращенной формы записи. Если задано одно значение свойства `padding`, оно применяется ко всем сторонам элемента, как показано в листинге 5.7.

Листинг 5.7. Код CSS, содержащий свойство `padding` с одним заданным значением (сокращенная форма записи)

```
p { padding: 1em; }
```

Если заданы два значения, величина нижнего и верхнего заполнения устанавливается равной первому, а величина левого и правого — равной второму значению, как показано в листинге 5.8.

Листинг 5.8. Код CSS, содержащий свойство `padding` с двумя заданными значениями (сокращенная форма записи)

```
p { padding: 1em 0; }
```

Если заданы три значения, величина верхнего заполнения устанавливается равной первому значению, левого и правого — второму, а величина нижнего заполнения — третьему, как показано в листинге 5.9.

Листинг 5.9. Код CSS, содержащий свойство `padding` с тремя заданными значениями (сокращенная форма записи)

```
p { padding: 1em 0 2em; }
```

Если заданы четыре значения, они определяют величину верхнего, правого, нижнего и левого заполнения, как показано в листинге 5.10.

Листинг 5.10. Код CSS, содержащий свойство `padding` с четырьмя заданными значениями (сокращенная форма записи)

```
p { padding: 1em 2em 2em 1em; }
```

Граница

Свойство `border` задает ширину, цвет и стиль границы вокруг элемента. В сокращенной форме записи используются такие свойства границы, как `border-top`, `border-bottom`, `border-right`, `border-left` и `border`, как показано в листинге 5.11.

Листинг 5.11. Код CSS, содержащий различные свойства `border` (сокращенная форма записи)

```
p { border-top: 1px solid red; }
p { border-right 1px solid red; }
p { border-bottom: 1px solid red; }
p { border-left: 1px solid red; }
p { border: 1px solid red; }
```

Область содержимого

Для области содержимого блока можно задать ширину, высоту и цвет. Ширину и высоту можно указывать в пунктах (1/72 дюйма), пиках (12 пунктов), пикселях, единицах em и ex, миллиметрах, сантиметрах, дюймах и процентах, как показано в листинге 5.12.

Листинг 5.12. Код CSS, содержащий различные значения `width` и `height`

```
p { width: 100pt; }
p { height: 20pc; }
p { width: 300px; }
p { height: 40em; }
p { width: 50ex; }
p { height: 600mm; }
p { width: 70cm; }
p { height: 8in; }
p { width: 50%; }
```

Для определения в стиле цвета текста используется свойство `color`. Цвет можно задавать по-разному: с помощью ключевых слов, шестнадцатеричных RGB-кодов и функциональной формы записи RGB.

В качестве ключевых слов свойство `color` воспринимает `aqua` (зеленовато-голубой), `black` (черный), `blue` (синий), `fuchsia` (светло-фиолетовый), `gray` (серый), `green` (зеленый), `lime` (светло-зеленый), `maroon` (коричневый), `navy` (темно-синий), `olive` (оливковый), `purple` (фиолетовый), `red` (красный), `silver` (светло-серый), `teal` (сине-зеленый), `white` (белый) и `yellow` (желтый). Хотя в некоторых браузерах могут использоваться и другие цвета, они не представлены в обязательной спецификации, поэтому их следует избегать.

Для шестнадцатеричного представления цветов можно использовать только три или шесть шестнадцатеричных символов, как показано в листинге 5.13. Если цвет представлен тремя парами шестнадцатеричных цифр (например, `#ff0000`), его можно представить в сокращенном виде, удалив по одной цифре из каждой пары (`#f00`). Чтобы задавать RGB-цвета, можно использовать три разделенных запятыми целых числа или значения компонентов в процентах. Например, красный цвет задается с помощью `rgb(255, 0, 0)` либо `rgb(100%, 0%, 0%)`, как показано в листинге 5.13.

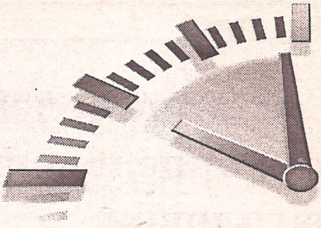
Листинг 5.13. Код CSS, содержащий различные значения свойства `color`

```
p { color: red }
p { color: #f00 }
p { color: #ff0000 }
p { color: rgb(255,0,0) }
p { color: rgb(100%, 0%, 0%) }
```

Резюме

В этом уроке вы ознакомились с блочной моделью CSS и ее компонентами — полями, цветом фона, фоновым изображением, заполнением и границей. Кроме того, вы узнали о разнице между внутритекстовыми и блочными элементами и о том, как некоторые версии Internet Explorer неверно интерпретируют блочную модель. В следующем уроке вы научитесь применять фоновое изображение к элементу <body>. Кроме того, вы узнаете, как использовать свойства `background-repeat` и `background-position`.

Урок 6



Добавление фоновых изображений

В этом уроке вы научитесь добавлять фоновое изображение к элементу <body>, узнаете, как применять свойства background-repeat и background-position. В качестве обучающей задачи мы будем создавать градиентное изображение, повторяющееся до правого нижнего угла страницы.

Подготовка HTML-кода

HTML-код этого урока будет состоять из трех абзацев текста, как показано в листинге 6.1.

Листинг 6.1. Код HTML, содержащий разметку для урока 6

```
<p>
    Lorem I
psum dolor sit amet...
</p>
<p>
    Ut wisi enim ad minim veniam...
</p>
<p>
    Duis autem vel eum iriure dolor...
</p>
```

Создание селекторов для определения стиля заголовка

Чтобы определить стиль элемента <body> и его содержимого, требуется единственный селектор типа, как показано в листинге 6.2.

Листинг 6.2. Код CSS, иллюстрирующий селектор, который используется для стилевого оформления абзаца

```
body {...}
```

Добавление фонового изображения

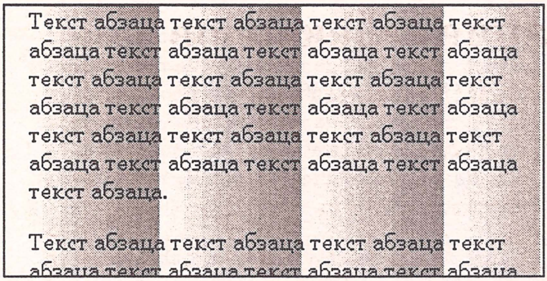
Для добавления фонового изображения к элементу <body> используется свойство `background-image`.

Значением свойства `background-image` может быть либо `url` (задается изображение), либо `none` (изображение не используется).

В данном примере мы будем использовать свойство `url(chapter6.jpg)`. Путь к изображению можно указывать как в кавычках, так и без них. Код свойства `background-image` показан в листинге 6.3. Результаты обработки этого кода представлены на рис. 6.1.

Листинг 6.3. Код CSS со стилевым оформлением элемента <body> (добавление фонового изображения)

```
body
{
    background-image: url(chapter6.jpg);
}
```



Текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца

Рис. 6.1. Стилевое оформление элемента <body>

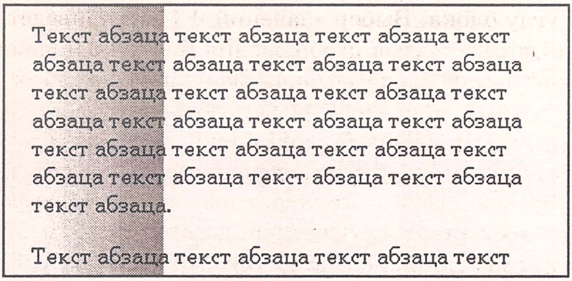
Фоновые изображения и Internet Explorer 5 для Macintosh

Internet Explorer 5 для Macintosh не визуализирует фоновые изображения, заданные с помощью `background-images`, если путь к изображению заключен в кавычки. Поэтому, поскольку эти символы не являются обязательными, проще и безопаснее их опускать.

Настройка свойства background-repeat

Сейчас фоновое изображение повторяется по всему экрану. Для управления этим повтором применяется свойство `background-repeat`.

В число допустимых значений свойства `background-repeat` (рис. 6.2) входят `repeat` (изображение повторяется по горизонтали и вертикали), `repeat-x` (изображение повторяется только по горизонтали), `repeat-y` (изображение повторяется только по вертикали) и `no-repeat` (изображение не повторяется).



Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст

Рис. 6.2. Стилевое оформление элемента <body> с заданным свойством background-repeat

Как показано в листинге 6.4, в этом уроке будет использоваться значение `repeat-y`, чтобы изображение повторялось по вертикали к нижнему краю страницы.

Листинг 6.4. Код CSS, содержащий настройки `background-repeat`

```
body
{
    background-image: url(chapter6.jpg);
    background-repeat: repeat-y;
}
```

Добавление свойства background-position

Теперь фоновое изображение повторяется, как нам требуется, и его нужно растянуть до правого нижнего края элемента `<body>`. Для этого применяется свойство `background-position`.

В качестве значений свойства `background-position` можно использовать значения в процентах (например, `0 100%`), длину (например, `2px 20px`) и ключевые слова (например, `left top`). В любом случае вначале задается горизонтальное положение, а затем вертикальное. Выбор значений `0% 0%` соответствует расположению левого верхнего угла изображения в левом верхнем углу блока. Выбор значений `0 100%` приведет к тому, что левый нижний угол изображения будет расположен в левом нижнем углу блока абзаца. Значения `2px 20px` зададут размещение левого верхнего угла изображения на расстоянии `2px` к центру и `20px` вниз от верха блока.

Если указано только одно значение в процентах или одна длина, устанавливается только горизонтальное расстояние; вертикальное в таком случае принимается равным `50%`. Если указаны два значения, первое из них соответствует расстоянию по горизонтали. Допускается смешанное использование значений в процентах и длин (например, `50% 2cm`). Кроме того, разрешены отрицательные значения (например, `-20px 10px`).

В данном уроке будут использованы значения в процентах `100% 0`, которые задают размещение изображения справа вверху элемента. Соответствующий код показан в листинге 6.5.

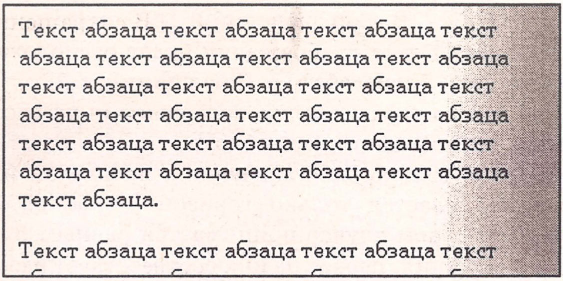
Теперь изображение будет повторяться вниз от правого края элемента `<body>` (рис. 6.3).

Вопросы, связанные с положением фона

Некоторые браузеры не распознают ключевое слово `right`, используемое в свойстве `background-position`. Тем не менее все современные браузеры правильно обрабатывают значение в процентах `100%`, поэтому рекомендуется использовать именно его.

Листинг 6.5. Код CSS со стилевым оформлением элемента `<body>` (добавлено свойство `background-position`)

```
body
{
    background-image: url(chapter6.jpg);
    background-repeat: repeat-y;
    background-position: 100% 0;
}
```



Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст

Рис. 6.3. Стилевое оформление элемента <body> с заданным свойством background-position

Использование сокращенной формы записи свойства background

Как обсуждалось в уроке 2, "Использование правил CSS", сокращенная форма записи свойств позволяет указывать значения нескольких свойств в одном. Благодаря этому в свойстве `background` можно объединить `background-color`, `background-image`, `background-repeat`, `background-attachment` и `background-position`.

При сортировке свойств в сокращенной форме записи браузеры вначале устанавливают все отдельные свойства согласно их исходным значениям, а затем переопределяют те из них, которые указал автор.

По умолчанию правило фона записывается как `background: transparent none repeat scroll 0 0;`. Если объединить использованные в этом уроке объявления в одно сокращенное правило, они переопределят значения, принятые по умолчанию для свойств `background-image`, `background-repeat` и `background-position`. В результате получим `background: transparent url(chapter6.jpg) repeat-y scroll 100% 0;`.

Тем не менее правило можно сократить еще больше, включив в него только необходимые объявления, так что оно примет вид `background: url(chapter6.jpg) repeat-y 100% 0;` (листинг 6.6).

Листинг 6.6. Код CSS со стилевым оформлением элемента `<body>` (сокращенная запись свойства `background`)

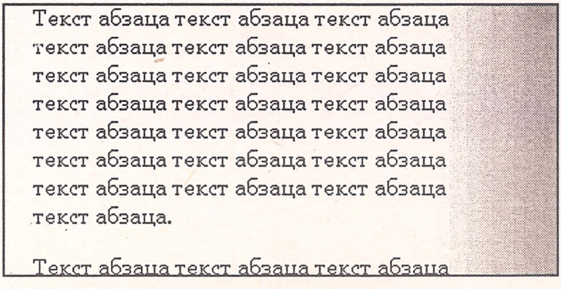
```
body
{
    background: url(chapter6.jpg) repeat-y 100% 0;
}
```

Добавление заполнения

Последний этап — добавление заполнения к элементу `<body>`, чтобы отделить текст от фонового изображения. Для этого можно использовать сокращенную форму записи `padding: 1em 80px 1em 1em;`, как показано в листинге 6.7. В результате вверху, внизу и слева от края элемента `<body>` будет размещено заполнение шириной `1em`, а справа ширина заполнения будет равна `80px`. Результаты выполнения соответствующего кода представлены на рис. 6.4.

Листинг 6.7. Код CSS с добавлением к элементу `<body>` заполнения

```
body
{
    background: url(chapter6.jpg) repeat-y 100% 0;
    margin: 0;
    padding: 1em 80px 1em 1em;
}
```



Текст абзаца текст абзаца.

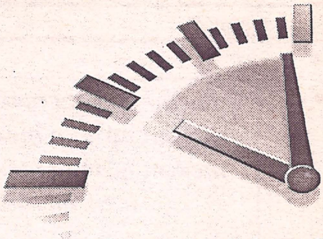
Текст абзаца текст абзаца текст абзаца

Рис. 6.4. Элемент <body> с добавленным заполнением

Резюме

В этом уроке вы научились применять фоновое изображение к элементу `<body>`. Кроме того, вы узнали, как использовать свойства `background-repeat`, `background-position` и сокращенную форму записи свойства `background`. В следующем уроке вы научитесь определять стилевое оформление текста с использованием свойств `font`, `size`, `color` и `alignment`.

Урок 7



Форматирование текста

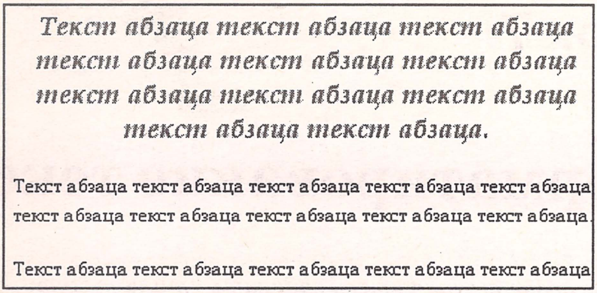
В этом уроке вы научитесь задавать стиль текста, используя вместо элемента `<font>` свойства `font`, `size`, `alignment` и `color`.

Подготовка HTML-кода

HTML-код этого урока содержит три абзаца текста, как показано в листинге 7.1. Содержимое этих абзацев помещено в элемент `<font>`. С первым абзацем сопоставлен шрифт большего размера. Кроме того, сам текст цветной, курсивный и полужирный. Внешний вид данного абзаца текста в браузере показан на рис. 7.1.

Листинг 7.1. Код HTML, содержащий разметку для урока 7

```
<p align="center">
    <font size="4" color="#990000" face="times, times new
    roman">
    <b><i>Lorem ipsum dolor sit amet, consectetuer adipiscing
    elit...</i></b></font>
</p>
<p>
    <font size="2" face="arial, helvetica">
    Ut wisi enim ad minim
veniam, quis nostrud exerci...</font>
</p>
<p>
    <font size="2" face="arial, helvetica">
    Duis autem vel eum iriure dolor in hendrerit vulputate...
    </font>
</p>
<p>
    <font size="2" face="arial, helvetica"><p> nostrud
    exerci...
    </font>
</p>
```



Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца

Рис. 7.1. *Абзацы, стиль которых задан с помощью элемента* `<font>`

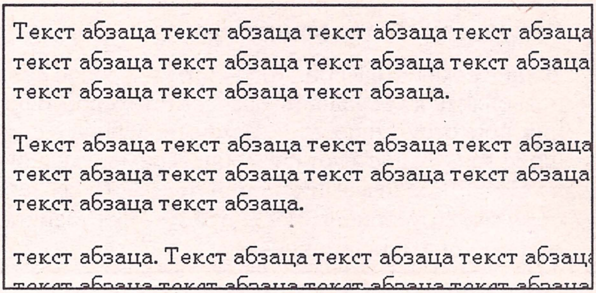
Удаление элементов `<font>`

Вместо использования в документе элементов `<font>` для задания стилевого оформления содержимого следует применять CSS. Это уменьшает общий размер файлов и облегчает управление ими в будущем. Всю информацию, относящуюся к стилю шрифтов, можно записать в одном внешнем файле, а не рассеивать по всем документам Web-сайта.

Элементы `<font>` можно удалить из разметки HTML, как показано в листинге 7.2. Поскольку стиль первого абзаца требует особого стилевого оформления, он будет определяться классом `introduction` (рис. 7.2).

Листинг 7.2. Код HTML, содержащий разметку вне элементов `<font>`

```
<p class="introduction">
    Lorem ipsum dolor sit amet, consectetuer adipiscing
    elit...
</p>
<p>
    Ut wisi enim ad minim veniam, quis nostrud exerci...
</p>
<p>
    Duis autem vel eum iriure dolor in hendrerit vulputate...
</p>
    Ut wisi enim ad minim veniam, quis nostrud exerci...
</p>
```



Текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца текст абзаца.

текст абзаца. Текст абзаца текст абзаца текст абзац

Рис. 7.2. Абзацы текста при удаленных элементах <font>

Создание селекторов

Для определения стиля абзацев будут использованы два селектора, как показано в листинге 7.3.

Листинг 7.3. Код CSS, иллюстрирующий селекторы, которые используются для определения стилей абзацев

```
p {...}
p.introduction {...}
```

Формирование стиля элемента <p>

Для задания семейства шрифтов используется свойство `font-family`. В этом свойстве всегда нужно указывать ряд шрифтов, разделенных запятыми. В конце списка следует указать общее семейство шрифтов. Если в системе пользователя нет семейства шрифтов, указанного первым, его браузер попытается использовать второе указанное семейство. Если подобным образом подходящее семейство не будет найдено, браузер использует общее семейство шрифтов.

Общие семейства шрифтов

Общие семейства шрифтов — это резервное средство, позволяющее создавать базовое стилевое оформление при отсутствии всех рекомендованных семейств шрифтов. Существует пять общих семейств шрифтов: `serif` (с засечками), `sans-serif` (без засечек), `cursive` (рукописный), `fantasy` (фантазийный) и `monospace` (моноширинный).

В нашем примере свойство `font-size` будет установлено равным `80%`, т.е. размер шрифта будет составлять 80 % от размера текущего стиля по умолчанию, определенного в браузере пользователя. С помощью величин в процентах пользователь может контролировать общий размер шрифтов (рис. 7.3).

Единицы `em` и проценты

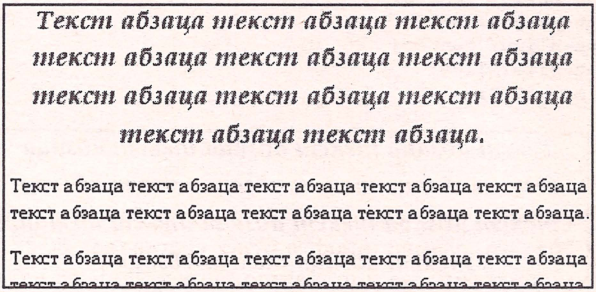
Теоретически задание размера шрифта с помощью единиц `em` или процентов ничем не отличается. Тем не менее Internet Explorer 5 для Windows неверно читает величины в единицах em, соответствующие размерам, меньшим 100 %, и заменяет единицы измерения em пикселями. Например, значение `.8em` будет отображено как `8px`.

Чтобы избежать данной проблемы, любые размеры шрифтов, меньшие 100 %, следует задавать в процентах.

Наконец, в код мы включаем значение `line-height`, равное `140%`, чтобы увеличить расстояние между строками и сделать текст более читабельным. В большинстве браузеров значение `line-height` по умолчанию равно `120%`. Задавая величину `140%`, мы добавляем 20 % дополнительного пустого места между строками. Соответствующий набор правил показан в листинге 7.4.

Листинг 7.4. Код CSS, содержащий стили для элемента `<p>`

```
p
{
    font-family: arial, helvetica, sans-serif;
    font-size: 80%;
    line-height: 140%;
}
```



Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца

Рис. 7.3. Абзацы с определенными стилями

Стилевое оформление первого абзаца

Первый абзац в нашем примере представлен стилем, отличным от стиля всех остальных абзацев. В таком случае его оформление будет задано как `times, "times new roman", serif`. Шрифты, подобно Times New Roman имеющие в названии пробелы, должны всегда заключаться в кавычки.

Следующий шаг — определить стиль как курсивный полужирный. Для этого применяются свойства `font-style: italic;` и `font-weight: bold;`.

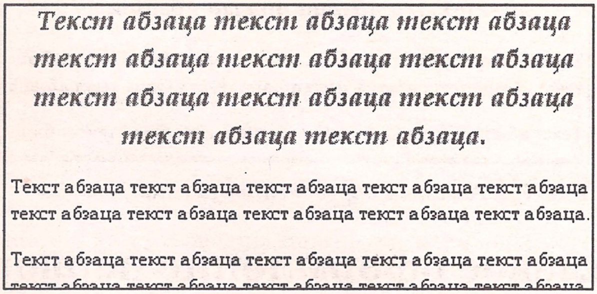
Чтобы выровнять текст по центру экрана, используется `text-align: center`.

Увеличить размер шрифта можно с помощью `font-size: 110%;`, а установить цвет — с помощью `color: #900;`, как показано в листинге 7.5 (рис. 7.4).

Листинг 7.5. Код CSS, содержащий стили для первого абзаца

```
p
{
    font-family: arial, helvetica, sans-serif;
    font-size: 80%;
    line-height: 1.4;
}
```

```
p.introduction
{
    font-family: times, "times new roman", serif;
    font-style: italic;
    font-weight: bold;
    text-align: center;
    font-size: 110%;
    color: #900;
}
```



Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца

Рис. 7.4. Первый абзац представлен требуемым стилем

Преобразование в сокращенную форму записи

Как обсуждалось в уроке 2, “Использование правил CSS”, сокращенная форма записи свойств облегчает написание и эксплуатацию кода. Кроме того, благодаря ей файлы CSS становятся более лаконичными.

Стиль элемента `<p>` можно определить так, чтобы `font-size`, `line-height` и `font-family` объявлялись, как одно свойство шрифта.

Стиль класса `introduction` можно определить так, чтобы `font-style`, `font-weight`, `font-size`, `line-height` и `font-family` объявлялись, как одно свойство шрифта, как показано в листинге 7.6.

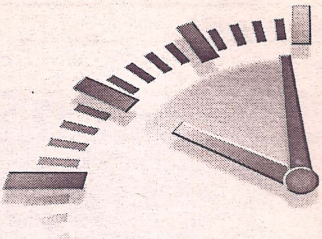
Листинг 7.6. Код CSS, содержащий стили в сокращенной форме записи

```
p
{
    font: 80%/1.4 arial, helvetica, sans-serif;
}
p.introduction
{
    font: bold italic 110%/1.4 times, "times new roman",
serif;
    text-align: center;
    color: #900;
}
```

Резюме

В этом уроке вы научились определять стили текста, используя свойства `font-family`, `fontsize`, `line-height`, `font-style`, `font-weight`, `text-align` и `color`. Кроме того, вы узнали, как использовать сокращенную запись свойст в шрифта. В следующем уроке вы научитесь определять стилевое оформление заголовков, используя свойства `border`, `background-images` и `text-transform`.

Урок 8



Стилевое оформление гибкого заголовка

В этом уроке вы научитесь создавать границу над и под заголовком, добавлять непрерывное градиентное фоновое изображение и определять стиль текста с помощью `text-transform` и разрядки.

Стилевое оформление заголовка

Чтобы определить стиль заголовка, потребуется селектор, выделяющий элемент `<h1>`. Чтобы гарантировать, что вы не затронете все элементы `<h1>` страницы, необходимо также включить в селектор уникальный идентификатор, например `header` (листинг 8.1).

Листинг 8.1. Код CSS, иллюстрирующий селектор, который используется для определения стиля заголовка

```
h1#header {...}
```

Код HTML, используемый для иллюстрации этого заголовка, показан в листинге 8.2.

Листинг 8.2. Код HTML, содержащий разметку для заголовка

```
<h1 id="header">
    Page Heading
</h1>
```

Уровни заголовков и структура документа

Для придания значения контексту Web-документы используют семантически правильную разметку. Например, заголовки нужно помещать внутрь элементов заголовков, абзацы текста — внутрь элементов абзацев, списки — внутрь элементов списков и т.д.

Если семантически правильная HTML-разметка имеется, для визуального изменения стиля содержимого можно использовать CSS.

Важной частью данной разметки являются уровни заголовков. В идеальном случае Web-страницы должны начинаться с единственного элемента `<h1>`, содержащего самую важную информацию на этой странице (например, название страницы или имя сайта).

Заголовки никогда нельзя имитировать, используя элементы `<font>` и `<strong>`, поскольку это не имеет смысла для программ чтения с экрана или текстовых браузеров.

Добавление цвета, размера шрифта и ширины

Чтобы добавить к заголовку текст, используйте свойство `color`. В любой момент вы можете заменить цвет более удобным для вас.

В данном примере свойство `font-size` задается равным `120%`, а `font-weight` — равным `normal`, как показано в листинге 8.3.

Листинг 8.3. Код CSS, задающий свойства `font-size` и `font-weight`

```
h1#header
{
    color: #036;
    font-size: 120%;
    font-weight: normal;
}
```

Переопределение стандартных настроек заголовков

Любой HTML-документ может иметь три или более соотнесенных с ним таблиц стилей, включая таблицу стилей *браузера*, *пользовательскую* таблицу стилей, а также одну или несколько *авторских* таблиц стилей.

Ко всем Web-документам браузеры применяют таблицы стилей по умолчанию. Хотя такие таблицы браузеров отличаются для разных браузеров, они обладают некоторыми общими характеристиками, например черный текст и синие ссылки.

Большинство современных браузеров позволяют пользователям устанавливать в браузере собственные таблицы стилей, которые имеют более высокий приоритет, чем таблицы браузера (но только для этого пользователя!).

Если вы применили к странице обычную таблицу стилей или внутритекстовый стиль, вы добавили авторскую таблицу стилей. Авторские таблицы стилей в общем случае имеют более высокий приоритет и по отношению к пользовательским стилям, и по отношению к стилям браузера.

Большинство таблиц стилей браузера отображает элемент `<h1>` полужирным шрифтом при 200 % размера шрифта по умолчанию. Если вы решите, что элемент `<h1>` должен иметь размер 120 %, будет использовано именно это значение, а не 200 %, поскольку ваши таблицы стилей имеют более высокий приоритет, чем таблицы стилей браузера.

Использование сокращенных шестнадцатеричных кодов цвета

Шестнадцатеричные коды цвета можно задавать с помощью символа #, за которым непосредственно следуют три или шесть шестнадцатеричных символов.

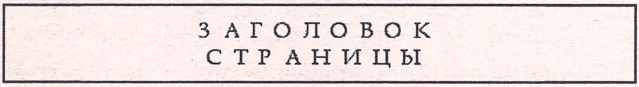
> Трехзначные шестнадцатеричные значения преобразуются в шестизначную форму путем дублирования цифр. Таким образом, `#f00` — это то же самое, что `#ff0000`, а `#f2a` — то же, что `#ff22aa`.

Задание опций текста

Следующий этап — отцентрировать заголовок, представить его прописными буквами и добавить разрядку. Все это можно сделать с помощью свойств `text-transform`, `text-align` и `letter-spacing`, как показано в листинге 8.4 и проиллюстрировано на рис. 8.1 (данные опции вы также можете изменить по своему усмотрению).

Листинг 8.4. Код CSS с преобразованием текста

```
h1#header
{
    color: #036;
    font-size: 120%;
    font-weight: normal;
    text-transform: uppercase;
    text-align: center;
    letter-spacing: .5em;
}
```



ЗАГОЛОВОК
СТРАНИЦЫ

Рис. 8.1. Заголовок отцентрирован и представлен прописными буквами

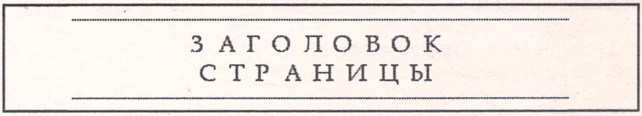
Добавление заполнения и границ

Позже в этом уроке вы будете добавлять границы сверху и снизу от заголовка. Чтобы текст не был расположен слишком близко к границам, необходимо небольшое заполнение сверху и снизу. Для его реализации можно использовать сокращенную форму записи свойства `padding`, задавая размер заполнения сверху и снизу равным `.4em`, а слева и справа — равным `0`.

Чтобы создать границы сверху и снизу от заголовка, используйте свойства `border-top` и `border-bottom`, как показано в листинге 8.5. Результаты выполнения данного кода представлены на рис. 8.2 (по желанию вы можете удалить или изменить данные границы).

Листинг 8.5. Код CSS, добавляющий границы

```
h1#header
{
    color: #036;
    font-size: 120%;
    font-weight: normal;
    text-transform: uppercase;
    text-align: center;
    letter-spacing: .5em;
    padding: .4em 0;
    border-top: 1px solid #069;
    border-bottom: 1px solid #069;
}
```



ЗАГОЛОВОК
СТРАНИЦЫ

Рис. 8.2. Заголовок с границами

Границы CSS

Свойство `border` определяет границы вокруг элемента.

- `border-color` устанавливает цвет границы (например, `red`, `transparent`, `none`, `#036`, `#003366`).
- `border-width` устанавливает толщину границы (например, `thin`, `medium`, `thick`, `1px`, `.5em`, `1ex`).
- `border-style` устанавливает внешний вид границы (например, `none`, `hidden`, `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset`, `outset`).

Свойства и значения границ можно задать множеством способов. Проще всего использовать сокращенную форму записи:

```
border: (width) (style) (color);
```

Некоторые стили границ, такие как `dotted`, не поддерживаются Internet Explorer 5 и 5.5.

Добавление фонового изображения

Чтобы добавить к заголовку фоновое изображение, используйте свойство `background`. После этого задайте указатель `url` и значение `repeat`, как показано в листинге 8.6. В данном случае изображение задается с параметром `repeat-x`, поэтому оно повторяется только по оси *x*. Результаты выполнения данного кода представлены на рис. 8.3. По желанию данное фоновое изображение вы можете убрать или изменить.

Листинг 8.6. Код CSS, добавляющий фоновое изображение

```
h1#header
{
    color: #036;
    font-size: 120%;
    font-weight: normal;
    text-transform: uppercase;
    text-align: center;
    letter-spacing: .5em;
    padding: .4em 0;
    border-top: 1px solid #069;
    border-bottom: 1px solid #069;
    background: url(chapter8.jpg) repeat-x;
}
```

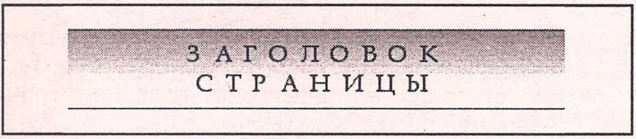
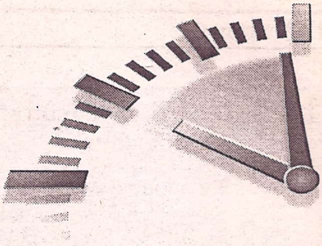


Рис. 8.3. Завершенный заголовок

Резюме

В этом уроке вы научились определять стиль гибкого заголовка, используя `font-size`, `font-weight`, границы, заполнение и фоновые изображения. В следующем уроке вы научитесь определять стиль заголовка с закругленными углами.

Урок 9



Заголовки в закругленной рамке

В этом уроке вы научитесь обрамлять заголовок закругленной рамкой. Рамка создается из двух фоновых изображений, растягивающихся до требуемого размера.

Определение стиля заголовка

Чтобы определить стиль заголовка, необходим селектор, выделяющий элемент <h2>. Чтобы гарантировать, что вы не затронете все элементы <h2> на странице, в селектор также необходимо включить класс. В данном случае мы используем именно класс, а не идентификатор, поскольку может потребоваться отметить на одной странице несколько заголовков (листинг 9.1).

Листинг 9.1. Код CSS, иллюстрирующий селекторы, которые используются для определения стиля заголовка

```
h2.decorative {...}
h2.decorative em {...}
```

Код HTML, используемый для иллюстрации, показан в листинге 9.2. Обратите внимание на то, что заголовок помещен в элемент <em> (“emphasis” — “выделение”). Этот дополнительный элемент мы объясним немного ниже.

Листинг 9.2. Код HTML, содержащий разметку для заголовка

```
<h2 class="decorative">
    <em>Section Heading</em>
</h2>
```

Создание масштабируемого фонового изображения

Заголовок, используемый в этом уроке, в конечном счете будет помещен в рамку с закругленными углами.

Эта рамка должна расширяться вниз, если текст заголовка окажется длинным или если пользователь решит выбрать в браузере больший размер шрифта.

Поэтому рамка делается из двух фоновых изображений. Первое из них представляет верхний фрагмент рамки, а второе — нижний.

Первое изображение должно быть очень длинным, чтобы при необходимости расти вниз.

Если первое фоновое изображение применяется к элементу `<h2>`, второе фоновое изображение должно применяться к другому элементу.

Один из простых выходов из этой ситуации заключается в помещении текста заголовка в элемент `<em>` и применении второго изображения именно к этому элементу. Поскольку второе изображение привязано к низу элемента `<em>`, текст заголовка может быть сколь угодно длинным.

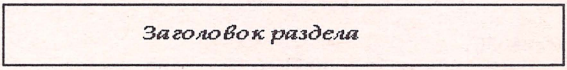
Стилевое оформление элемента <h2>

Чтобы добавить цвет к элементу <h2>, необходимо использовать свойство `color` (если вам не нравится наш выбор, можете изменить цвет).

Ширина, размер и семейство шрифта задаются с использованием свойства `font`, заголовок центрируется с помощью свойства `text-align`, как показано в листинге 9.3 и иллюстрируется на рис. 9.1.

Листинг 9.3. Код CSS, устанавливающий выравнивание текста

```
h1#header
{
    color: #036;
    font: bold 100% arial, helvetica, sans-serif;
    text-align: center;
}
```



Заголовок раздела

Рис. 9.1. Заголовок с переопределенным стилем

Добавление фонового изображения

Чтобы добавить к элементу `<h2>` фоновое изображение, используйте свойство `background`. Изображение следует задавать с параметром `no-repeat`, чтобы оно не продублировалось в середине длинного заголовка. Изображение, используемое нами в качестве фонового, показано на рис. 9.2.

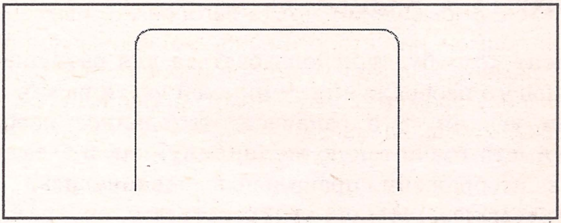


Рис. 9.2. Первое изображение, применяемое к элементу <h2>

Кроме того, необходимо задать ширину (`width`) этого элемента, поскольку фоновое изображение имеет размер 220 пикселей. Если ширина не определена слева, заголовок будет выходить за пределы рамки.

Кроме того, к верхней стороне элемента необходимо применить заполнение величиной 5 пикселей, как показано в листинге 9.4 и иллюстрируется на рис. 9.3. Данное заполнение немного опустит текст, так что он не будет вплотную примыкать к фоновому изображению.

Листинг 9.4. Код CSS, устанавливающий фоновое изображение, ширину и заполнение

```
h1#header
{
    color: #036;
    font: bold 100% arial, helvetica, sans-serif;
    text-align: center;
    background: url(chapter9.gif) no-repeat;
    width: 220px;
    padding: 5px 0 0 0;
}
```

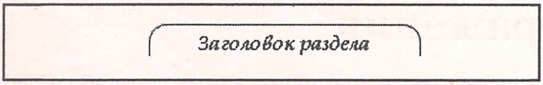


Рис. 9.3. Заголовок с фоновым изображением

Стилевое оформление элемента `<em>`

Элемент `<em>` будет использоваться для размещения второго фонового изображения — нижней части рамки с закругленными углами. Это означает, что данное изображение должно иметь точно такую же ширину, как и элемент `<h2>`, чтобы два изображения правильно выравнивались.

Тем не менее, поскольку `<em>` — это внутритекстовый элемент, он будет игнорировать любую заданную ширину. Решение заключается в присвоении ему значения `display: block;` перед применением параметра `width`.

Далее используется свойство `padding-bottom`, чтобы немного поднять текст относительно фонового изображения.

Кроме того, можно переопределить принятый по умолчанию курсивный стиль элемента `<em>`, использовав `font-style: normal;`.

Наконец необходимо применить фоновое изображение, используя свойство `background`. Изображение должно задаваться с параметром `no-repeat`, чтобы оно не повторялось под изображением. Положение фона нужно задать равным `0 100%`, чтобы левый нижний угол изображения совпадал с левым нижним углом элемента.

Получающееся в результате изображение показано на рис. 9.4, а соответствующий код CSS представлен в листинге 9.5. Полностью заголовок показан на рис. 9.5.

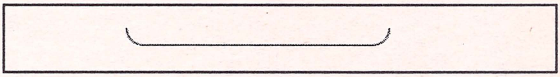


Рис. 9.4. Второе изображение, применяемое к элементу `<em>`

Листинг 9.5. Код CSS со стилевым оформлением элемента `<em>`

```
h1#header
{
    color: #036;
    font: bold 100% arial, helvetica, sans-serif;
    text-align: center;
    background: url(chapter9.gif) no-repeat;
    width: 220px;
    padding: 5px 0 0 0;
}
h2.decorative em
{
    display: block;
    width: 220px;
    padding: 0 0 5px 0;
    font-style: normal;
    background: url(chapter9a.gif) no-repeat 0 100%;
}
```

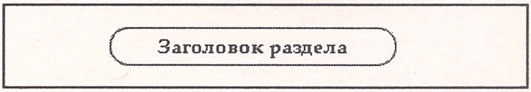
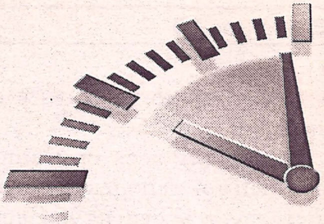


Рис. 9.5. Окончательный вид заголовка

Резюме

В этом уроке вы научились применять к заголовку цвет, шрифт и выравнивание текста. Кроме того, вы узнали, как задать фоновые изображения для двух элементов, чтобы получить рамку с закругленными углами. Из следующего урока вы узнаете о стилевом оформлении ссылок.

Урок 10



Стилевое оформление ссылок

Из этого урока вы узнаете о стилевом оформлении ссылок. Кроме того, вы научитесь добавлять к ссылкам фоновые изображения, отключать подчеркивание ссылок, использовать границы для подчеркивания и увеличивать активную область ссылки.

Ссылки и псевдоклассы

Вы уже знаете, как описывать стиль элемента <a> (элемент ссылки), — об этом рассказывалось в уроке 3, "Селекторы в действии". Теперь вы узнаете о стилевом оформлении пяти различных состояний ссылки.

Итак, ссылки могут находиться в следующих состояниях.

- **Обычное.** Стандартное состояние непосещенной ссылки.
- **Посещенное.** Ссылка указывает на идентификатор URI, который уже посещался.
- **Наведенное.** Курсор находится над активной областью ссылки.
- **Активное.** Момент выбора ссылки или щелчка на ней.
- **В фокусе.** Ссылка находится в фокусе и готова принимать вводимую информацию (например, щелчок мышью).

Некоторые состояния ссылки не могут существовать одновременно. Например, ссылка может быть или посещенной, или непосещенной. Тем не менее посещенные и непосещенные ссылки могут быть в наведенном, активном состоянии в фокусе.

Стили для всех описанных состояний можно определять независимо, используя псевдокласс `link` (класс, который не существует в структуре документа). Возможны следующие пять псевдоклассов:

- `a:link` — стили элементов непосещенных ссылок;
- `a:visited` — стили элементов посещенных ссылок;
- `a:focus` — стили состояния в фокусе;
- `a:hover` — стили состояния при установке курсора на ссылку;
- `a:active` — стили состояния при активизации ссылки.

Описанные пять псевдоклассов показаны в листинге 10.1.

Листинг 10.1. Код CSS, содержащий пять псевдоклассов ссылок

```
a:link {...}
a:visited {...}
a:focus {...}
a:hover {...}
a:active {...}
```

Состояние в фокусе и активное состояние

Псевдокласс `a:focus` подсвечивает положение табуляции для пользователей, для навигации применяющих клавиатуру.

К сожалению, Internet Explorer для Windows не поддерживает псевдокласс `a:focus`. Вместо этого в данном браузере для подсветки табуляции используется псевдокласс `a:active`.

Мало того, Internet Explorer для Windows неправильно применяет псевдокласс `a:active`. Состояние `a:active` остается видимым до тех пор, пока не будет предпринято другое действие.

Установка порядка следования псевдоклассов

Пять псевдоклассов имеют одинаковый вес, поэтому порядок, в котором они размещаются в файле CSS, имеет большое значение. Объявления псевдоклассов, которые появляются в файле CSS позже, заменяют собой предыдущие. Правильный порядок показан в листинге 10.2.

Листинг 10.2. Код CSS, содержащий правильный порядок следования псевдоклассов <a>

```
a {...}
a:link {...}
a:visited {...}
a:focus {...}
a:hover {...}
a:active {...}
```

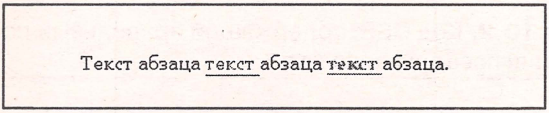
Использование классов с псевдоклассами

С псевдоклассами можно объединять селекторы класса, создавая ссылки для различных целей. Например, можно организовать стилевое оформление ссылок в зависимости от того, являются они внутренними или внешними.

Как показано в листинге 10.3, ко всем внешним ссылкам можно добавить класс; затем для этих ссылок можно создать стиль, используя объединенный селектор класса и псевдокласса. Результаты выполнения данного кода представлены на рис. 10.1.

Листинг 10.3. Код CSS, демонстрирующий объединенный селектор класса и псевдокласса

```
a:link
{
    color: blue;
}
a:visited
{
    color: purple;
}
a.external:link
{
    color: red;
    font-weight: bold;
}
a.external:visited
{
    color: black;
    font-weight: bold;
}
```



Текст абзаца текст абзаца текст абзаца.

Рис. 10.1. Разница между `a:link` *и* `a.external:link`

Стилевое оформление ссылок с фоновыми изображениями

Развивая идеи, использованные в предыдущем примере, можно создать небольшую пиктограмму за каждой внешней ссылкой.

Первый этап реализации этой идеи состоит в создании стандартных правил для требуемых состояний ссылки. В данном случае можно использовать `a:link`, `a:visited` и `a:hover`.

Затем добавляются три новых состояния (для ссылок, стиль которых задан с помощью “внешнего” класса): `a.external:link`, `a.external:visited` и `a.external:hover`.

В каждом случае к ссылке добавляется фоновое изображение. Изображение устанавливается с параметром `no-repeat`, поэтому оно не повторяется по фону всей ссылки. Положение устанавливается равным `100%`, т.е. правый край изображения сопоставляется с правым краем ссылки.

Для всех трех состояний используется одно фоновое изображение. Для каждого из этих состояний необходимо изменить вертикальный параметр `background-position`. Это означает, что загружается и заносится в кэш одно изображение, поэтому, если требуется изображение для реализации визуального выделения, никакой задержки не наблюдается.

Состояние `a:link` устанавливается равным 0; состояние `a:visited` — равным –100px; состояние `a:hover` — равным –200px. Фоновое изображение показано на рис. 10.2.

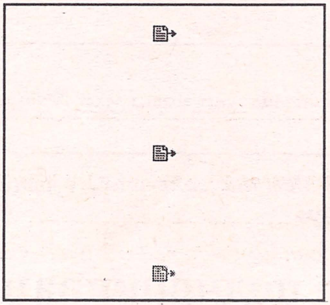


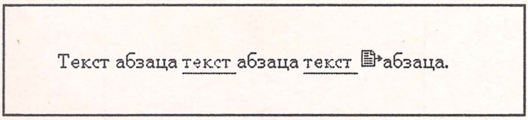
Рис. 10.2. Фоновое изображение, используемое в стилевом оформлении ссылки `.external`

Наконец, чтобы отодвинуть содержимое ссылки от фонового изображения, используется заполнение, как показано в листинге 10.4 (рис. 10.3).

Листинг 10.4. Код CSS, содержащий разметку для стилевого оформления ссылки `.external`

```
a:link
{
    color: blue;
}
a:visited
{
    color: purple;
}
a:hover
{
    color: red;
}
a.external:link
{
    background: url(chapter10.gif) no-repeat 100% 0;
    padding-right: 20px;
}
a.external:visited
```

```
{
    background: url(chapter10.gif) no-repeat 100% -100px;
    padding-right: 20px;
}
a.external:hover
{
    background: url(chapter10.gif) no-repeat 100% -200px;
    padding-right: 20px;
}
```



Текст абзаца текст абзаца текст абзаца.

Рис. 10.3. Ссылка `.external` *с фоновым изображением*

Удаление подчеркивания и применение границ

Для некоторых пользователей (особенно имеющих плохое зрение) стандартное подчеркивание ссылки может оказаться трудночитаемым. Особенно это справедливо для ссылок, содержащих курсивный текст.

Одним из решений данной проблемы является отключение подчеркивания текста и использование границ.

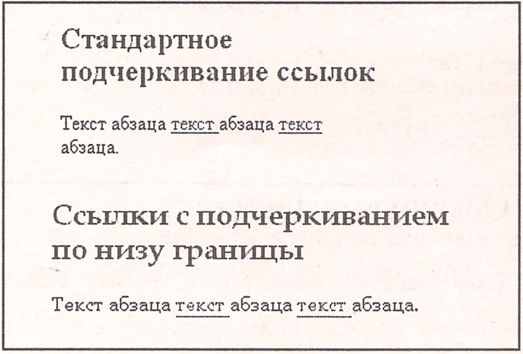
Первый шаг заключается в установке значения `text-decoration` равным `none`. Это отключает подчеркивание ссылок.

Далее необходимо раскрасить состояния. В данном случае цвет `a:link` устанавливается синим, `a:visited` — фиолетовым, а `a:hover` — красным.

Наконец к каждому состоянию с помощью `border-bottom` добавляются границы, как показано в листинге 10.5. Чтобы контролировать расстояние между подчеркиванием и содержимым (если это требуется), можно добавить свойство `padding-bottom` (рис. 10.4).

Листинг 10.5. Код CSS, устанавливающий стилевое оформление ссылок с границами

```
a
{
    text-decoration: none;
}
a:link
{
    color: blue;
    border-bottom: 1px solid blue;
}
a:visited
{
    color: purple;
    border-bottom: 1px solid purple;
}
a:hover
{
    color: red;
    border-bottom: 1px solid red;
}
```



Стандартное подчеркивание ссылок

Текст абзаца текст абзаца текст абзаца.

Ссылки с подчеркиванием по низу границы

Текст абзаца текст абзаца текст абзаца.

Рис. 10.4. Разница между ссылками с подчеркиванием и ссылками с заданным свойством `border-bottom`

Увеличение активной области ссылок

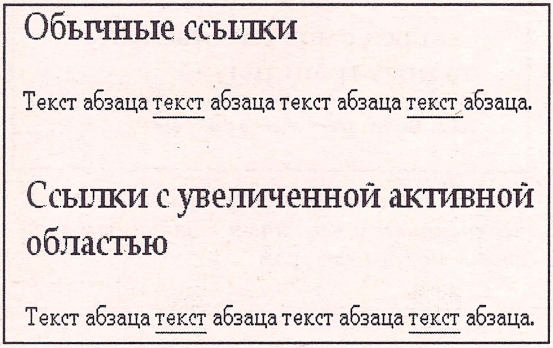
Некоторым пользователям (особенно с нарушенными моторными навыками) может быть трудно щелкать на ссылке. С помощью CSS активную площадь ссылок можно увеличить.

Для начала необходимо добавить `.5em` заполнения над и под элементом `<a>`, чтобы увеличить активную область ссылки. Для этого используется команда `padding: .5em 0;`.

Далее элемент `<a>` необходимо установить со свойством `position: relative`, что не позволит заполнению “переползти” на окружающий текст, как показано в листинге 10.6 (рис. 10.5).

Листинг 10.6. Код CSS, содержащий стили, увеличивающие активную область ссылок

```
a
{
    padding: .4em 0;
    position:relative;
    z-index: 1;
    background: yellow;
}
```



Обычные ссылки

Текст абзаца текст абзаца текст абзаца текст абзаца.

Ссылки с увеличенной активной областью

Текст абзаца текст абзаца текст абзаца текст абзаца.

Рис. 10.5. Разница между стандартной областью ссылки и увеличенной активной областью

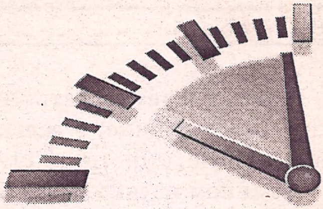
Чтобы визуализировать увеличенную область ссылки, для элемента `<a>` можно задать цвет фона. Перед использованием в реальной ситуации этот цвет удаляется.

Более подробное объяснение данной техники можно найти на Web-сайте Дэвида Бентона (David Benton) по адресу `http://www.dbenton.com/go/chronicles/2004/08/22/fitts-law-and-text-links/`.

Резюме

Из этого урока вы узнали, как создавать стили для ссылок и псевдоклассов. Вы научились применять фоновые изображения и границы к элементу `<a>`. Кроме того, вы узнали, как увеличить активную область ссылок, чтобы сделать их более доступными. В следующем уроке вы научитесь размещать и применять стилевое оформление к изображению и его подписи.

Урок 11



Расположение изображения и его подписи

Из этого урока вы узнаете, как размещать изображение и его подпись и задавать для них стили.

Оболочка для изображения и подписи

Первое, что нам необходимо сделать, — создать вокруг изображения и подписи контейнер-оболочку, чтобы эту пару элементов можно было переносить вместе. Для решения задачи можно использовать множество элементов, например абзацы, списки и даже списки определений. Тем не менее в данном уроке мы будем использовать `<div>`, как показано в листинге 11.1.

Листинг 11.1. Код HTML, содержащий разметку для контейнера, изображения и его подписи

```
<div class="imagecaption">
    <img src="chapter11.gif" alt="">
    A flower from my garden.
</div>
<p>
    Lorem ipsum dolor sit amet...
</p>
```

Чтобы гарантировать, что вы не захватите все элементы `<div>` на странице, в селектор необходимо включить класс, как показано в листинге 11.2.

В данном случае мы используем именно класс, а не идентификатор, поскольку на странице может обрабатываться несколько “плавающих” изображений с подписями.

Листинг 11.2. Код CSS, иллюстрирующий селекторы для стилевого оформления контейнера

```
div.imagecaption {...}
div.imagecaption img {...}
```

“Плавающий” контейнер

Чтобы переместить контейнер вдоль правого края окна браузера, следует использовать свойство `float: right`.

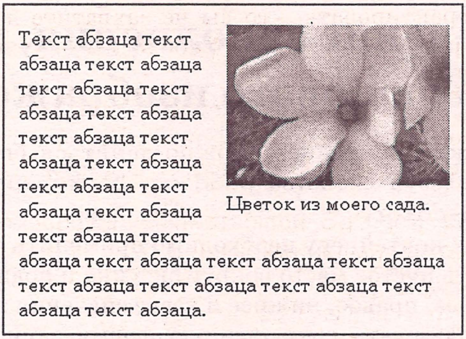
Для “плавающего” контейнера необходимо задать ширину. В данном случае используется свойство `width: 182px`.

Почему ширина такая странная? Изображение внутри данного контейнера имеет ширину `180px`. Ниже в этом уроке для изображения будет задана граница шириной `1px`. Ширина правой границы (`1px`), ширина левой границы (`1px`) и ширина изображения (`180px`) в сумме дают `182px`.

Далее можно создать немного пустого места вокруг контейнера, чтобы текст и другие элементы не слипались (рис. 11.1). Для этого можно указать правое, нижнее и левое поля контейнера, как показано в листинге 11.3.

Листинг 11.3. Код CSS, определяющий “плавающий” контейнер

```
div.imagecaption
{
    float: right;
    width: 182px;
    margin: 0 1em 1em 1em;
    display: inline;
}
```



Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Цветок из моего сада.

Рис. 11.1. "Плавающий" контейнер

"Плавающие" объекты, поля и Internet Explorer 5

Если вы просматриваете примеры из этого урока в Internet Explorer 5 или 5.5 для Windows, вы заметите, что правое поле в действительности гораздо шире, чем в других браузерах. Фактически его ширина равна `2em` — удвоенной предполагаемой ширине. Данный эффект называется *удвоением поля плавающего объекта в Internet Explorer*.

Данная ошибка проявляется, когда вы применяете правое поле к правому "плавающему" элементу и оно располагается непосредственно рядом с правым краем родительского контейнера.

Обратное также справедливо. Дефект наблюдается, когда вы применяете левое поле к левому "плавающему" элементу и оно располагается рядом с левым краем родительского контейнера.

К счастью, существует решение этой проблемы. Просто добавьте в набор правил `display: inline`. Все остальные браузеры проигнорируют это объявление, а Internet Explorer 5 и 5.5 для Windows будут применять правильную ширину поля.

Применение заполнения, цвета фона и фонового изображения

Расположив контейнер, можно заняться его внешним видом, используя свойства `padding`, `background-image` и `background-color`.

Сначала к контейнеру необходимо применить заполнение, чтобы создать пустое место вокруг рисунка и подписи. Чтобы задать верхнее, правое, нижнее и левое значения заполнения, можно использовать сокращенную форму записи свойства `padding`. Всем сторонам должно быть присвоено значение `10px`, кроме верхней, которой присвоено значение `70px`. В результате получится пустое пространство вокруг изображения.

Фоновое изображение можно задать с помощью `background-image: url(chapter11.gif)`, как показано на рис. 11.2.

Чтобы изображение не повторялось по всей фоновой области контейнера, необходимо правильно установить режим повтора изображения. В нашем случае изображение должно повторяться по оси *x*, поэтому используется свойство `background-repeat: repeat-x`.

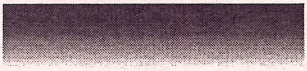


Рис. 11.2. Фоновое изображение

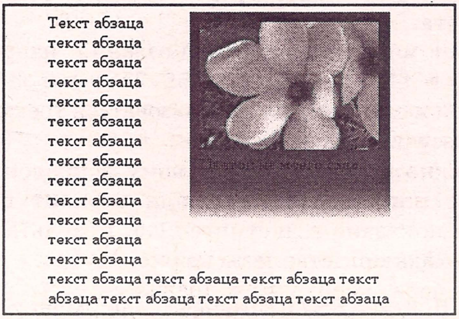
Фоновое изображение необходимо выровнять по нижней границе контейнера. Для этого можно присвоить свойству `vertical` значение `100%`, при этом нижняя граница изображения будет помещена рядом с нижней границей области заполнения. Соответствующее объявление выглядит как `background-position: 0 100%`.

Цвет фона можно задать с помощью свойства `background-color: #036`.

Все названные объявления фона можно собрать в одно, как показано в листинге 11.4. Результаты выполнения данного кода иллюстрируются на рис. 11.3.

Листинг 11.4. Код CSS, устанавливающий заполнение, цвет фона и фоновое изображение

```
div.imagecaption
{
    float: right;
    width: 182px;
    margin: 0 1em 1em 1em;
    display: inline;
    padding: 10px 10px 70px 10px;
    background: #036 url(chapter11.gif) repeat-x 0 100%;
}
```



Текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца текст абзаца текст абзаца текст
абзаца текст абзаца текст абзаца текст абзаца

Рис. 11.3. Контейнер с заполнением и фоном

Ключевые слова положения фона

При размещении фонового изображения можно использовать линейные размеры (например, `1em`, `20%` и `5px`) или ключевые слова (`left`, `right`, `center`, `top` и `bottom`).

К сожалению, некоторые версии Internet Explorer и Opera для Windows игнорируют ключевые слова.

К счастью, вместо них можно использовать значения в процентах, которые дают точно такой же эффект.

Если вам требуется расположить фоновое изображение справа от контейнера, можете использовать значение горизонтального смещения, равное `100%`. Это приведет к выравниванию правого края изображения по правому краю контейнера.

> Если необходимо расположить фоновое изображение внизу контейнера, можно использовать значение вертикального смещения, равное `100%`. Это приведет к выравниванию нижнего края изображения по нижнему краю контейнера.

Стилевое оформление подписи

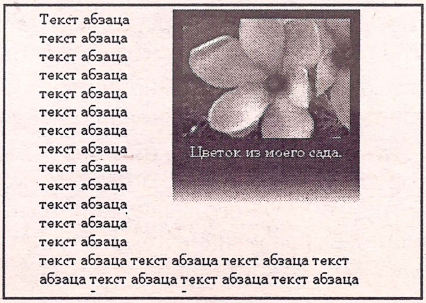
Следующий шаг — применить стили к тексту подписи. Начнем с цвета.

Существует множество способов задания, например, белого цвета: `#fff`, `#ffffff`, `rgb(255,255,255)` и `rgb(100%,100%,100%)`. В нашем случае будет использоваться трехзначное шестнадцатеричное представление `color: #fff`.

Текст можно выровнять по левому и правому краям, по центру и по ширине. Мы выбрали вариант `text-align: center`, как показано в листинге 11.5. Результаты выполнения данного кода представлены на рис. 11.4.

Листинг 11.5. Код CSS со стилевым оформлением подписи

```
div.imagecaption
{
    float: right;
    width: 182px;
    margin: 0 1em 1em 1em;
    display: inline;
    padding: 10px 10px 70px 10px;
    background: #036 url(chapter11.gif) repeat-x 0 100%;
    color: #fff;
    text-align: center;
}
```



Текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца текст абзаца текст абзаца текст
абзаца текст абзаца текст абзаца текст абзаца

Цветок из моего сада.

Рис. 11.4. Стилевое оформление подписи

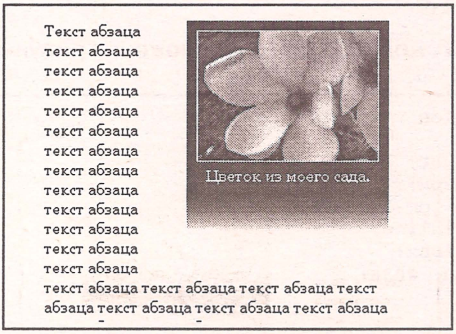
Стилевое оформление изображения

Изображение можно снабдить границей.

Существует множество способов задания рамки вокруг изображения. Простейший из них — использовать сокращенную форму записи свойства `border`, показанную в листинге 11.6 и проиллюстрированную на рис. 11.5.

Листинг 11.6. Код CSS со стилевым оформлением изображения

```
div.imagecaption
{
    float: right;
    width: 182px;
    margin: 0 1em 1em 1em;
    display: inline;
    padding: 10px 10px 70px 10px;
    background: #036 url(chapter11.gif) repeat-x 0 100%;
    color: #fff;
    text-align: center;
}
div.imagecaption img
{
    border: 1px solid #fff;
}
```



Текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца

Цветок из моего сада.

Рис. 11.5. Конечный результат

Расположение подписи рядом с рисунком

С помощью одних и тех же селекторов и кода HTML можно изменить структуру страницы, чтобы изображение и подпись располагались рядом.

Прежде всего, необходимо увеличить ширину контейнера, чтобы он вмещал новое положение подписи и изображения. Меняем объявление на `width: 302px`. Такой ширины должно быть достаточно.

Далее устанавливаем заполнение `10px` для всех сторон, поскольку уже не требуется место для фонового изображения.

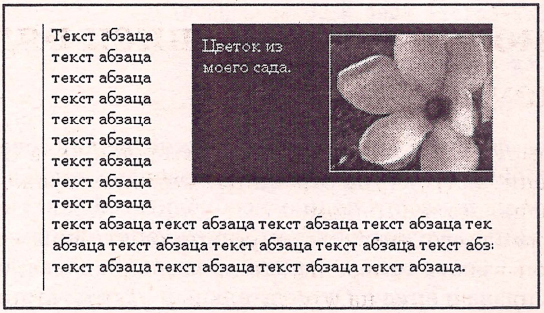
Свойства `background-image`, `background-repeat` и `background-position` также можно удалить, оставив единственное объявление `background: #036`.

Изображение следует прижать к правой стороне, чтобы подпись могла расположиться рядом с ним. В данном случае ширину определять уже не нужно, поскольку изображение имеет собственную ширину.

Наконец для изображения необходимо определить поля, чтобы подпись не вторглась в изображение. Для этого можно использовать свойство `margin-right: 1em`. Получающийся результат показан в листинге 11.7 и проиллюстрирован на рис. 11.6.

Листинг 11.7. Код CSS для расположения рисунка и подписи рядом

```
div.imagecaption
{
    float: right;
    width: 302px;
    margin: 0 1em 1em 1em;
    display: inline;
    padding: 10px;
    background: #036;
    color: #fff;
}
div.imagecaption img
{
    float: right;
    margin-left: 1em;
    border: 1px solid #fff;
}
```



Текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца
текст абзаца текст абзаца текст абзаца текст абзаца тек
абзаца текст абзаца текст абзаца текст абзаца текст абз
текст абзаца текст абзаца текст абзаца текст абзаца.

Цветок из
моего сада.

Рис. 11.6. Рисунок и подпись расположены рядом

Создание фотоснимка

Другим вариантом является создание простого фотоснимка с использованием контейнера и границ.

Первый этап — изменить заполнение на `15px` для верхней, правой и левой сторон и на `20px` для нижней стороны. Как и в первоначальном примере, дополнительное место будет использовано для фонового изображения.

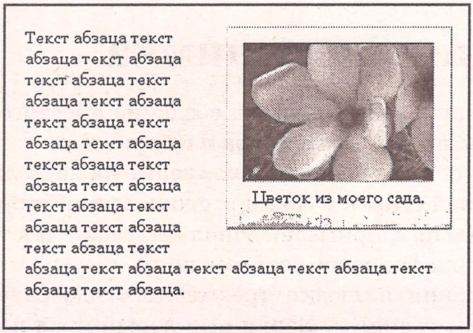
Для создания иллюзии трехмерного изображения будут использованы границы. Верхний и левый края получат светлые тонкие границы, а правый и нижний — темные и толстые. Для реализации сказанного можно использовать три объявления границ, как показано в листинге 11.8.

Листинг 11.8. Код CSS для создания фотоснимка

```
div.imagecaption
{
    float: right;
    width: 182px;
    margin: 0 1em 1em 1em;
    padding: 15px 15px 20px 15px;
    display: inline;
    text-align: center;
    border-color: #CCC #999 #999 #CCC;
    border-width: 1px 2px 2px 1px;
    border-style: solid;
    background: url(chapter11c.gif) repeat-x 0 100%;
}
div.imagecaption img
{
    border-color: #000 #ccc #ccc #000;
    border-width: 1px 1px 1px 1px;
    border-style: solid;
}
}
```

Новое фоновое изображение можно использовать с теми же настройками, которые были в первоначальной версии.

Наконец к изображению необходимо добавить границы. Эти границы рисуются цветом, противоположным цвету контейнера, — более темные границы вверху и слева, более светлые — справа и внизу. Окончательный результат представлен на рис. 11.7.



Текст абзаца текст абзаца.

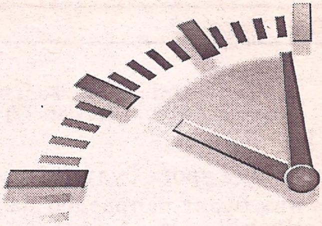
Цветок из моего сада.

Рис. 11.7. Фотоснимок

Резюме

Из этого урока вы узнали, как создавать контейнер-оболочку вокруг изображения и его подписи. После этого вы научились создавать “плавающий” контейнер, определять его стиль с использованием ширины, полей, заполнения, фонового изображения, цвета и выравнивания текста. Кроме того, вы узнали, как применять границы к изображению. В следующем уроке вы научитесь создавать фотогалерею.

Урок 12



Создание фотогалереи

В этом уроке вы научитесь создавать фотогалерею, используя ряд “плавающих” элементов <div>. Кроме того, вы узнаете, как использовать два фоновых изображения для создания гибкого контейнера.

Создание галереи с эскизами

Прежде всего, требуется ряд изображений-эскизов и подписей. Каждое изображение и подпись необходимо поместить внутрь элемента `<div>`. После этого подпись будет помещена внутрь элемента `<p>`, как показано в листинге 12.1.

Чтобы гарантировать, что вы не захватите все элементы `<div>` на странице, со всеми нужными элементами необходимо сопоставить одно имя класса.

Листинг 12.1. Код HTML, содержащий разметку для галереи эскизов

```
<div class="thumbnail">
    <img src="chapter12c.jpg" alt="">
    <p>A flower from my garden</p>
</div>
<div class="thumbnail">
    <img src="chapter12c.jpg" alt="">
    <p>White and pinkflower in Spring</p>
</div>
<div class="thumbnail">
    <img src="chapter12c.jpg" alt="">
    <p>Flower in morning light</p>
</div>
<div class="thumbnail">
    <img src="chapter12c.jpg" alt="">
    <p>A close-up of flower petals</p>
</div>
<div class="thumbnail">
```

```
    <img src="chapter12c.jpg" alt="">
    <p>A timeless flower </p>
</div>
```

В этом уроке будут использоваться три селектора. Первый из них будет направлен на все элементы `<div>`, содержащие класс `"thumbnail"`.

Второй селектор будет направлен на все изображения внутри элемента `<div>`, содержащего класс `"thumbnail"`.

Третий селектор будет направлен на все элементы `<p>` внутри элемента `<div>`, содержащего класс `"thumbnail"`. Данные селекторы показаны в листинге 12.2.

Листинг 12.2. Код CSS, иллюстрирующий селекторы, которые используются в стилевом оформлении контейнера

```
div.thumbnail {...}
div.thumbnail img {...}
div.thumbnail p {...}
```

Расположение элементов `<div>`

Поскольку изображения и подписи будут располагаться рядом, требуется "плавающий" элемент `<div>`. Для этого можно использовать свойство `float: left`.

Если элемент `<div>` является "плавающим", для него необходимо задать ширину. В нашем случае используется свойство `width: 130px` (вы можете выбрать любое удобное для вас значение).

Далее можно создать пустое пространство вокруг элементов `<div>`, чтобы они не перекрывали друг друга. Для этого к верхней и нижней границам каждого элемента `<div>` можно прикрепить поля.

Наконец, к `<div>` будет добавлено фоновое изображение, показанное на рис. 12.1. Как видно на рисунке, изображение очень длинное, поэтому оно может "расти", вмещая большие подписи.

Указанное фоновое изображение следует задать с параметром `no-repeat`, поскольку вам не нужно, чтобы оно повторно появилось под подписью. Соответствующий код CSS показан в листинге 12.3 и проиллюстрирован на рис. 12.2.

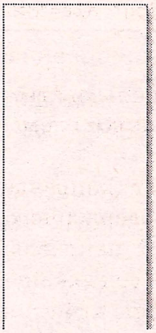


Рис. 12.1. Фоновое изображение, используемое элементом `<div>`

Листинг 12.3. Код CSS, представляющий “плавающий” контейнер

```
div.thumbnail
{
    width: 130px;
    float: left;
    margin: 0 10px 10px 0;
    background: url(chapter12a.gif) no-repeat;
}
```

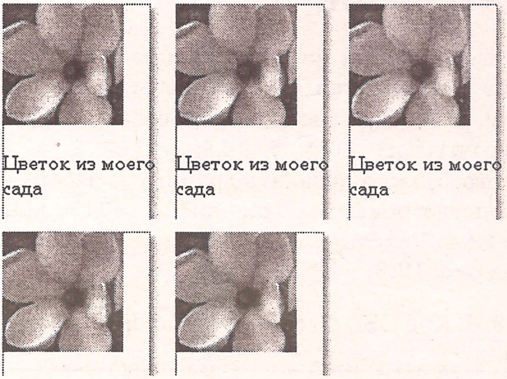

Цветок из моего сада
Цветок из моего сада
Цветок из моего сада

Рис. 12.2. Размещенные контейнеры

“Текучая” и фиксированная структуры

Созданные вами “плавающие” элементы `<div>` будут располагаться рядом в зависимости от ширины окна вашего браузера.

Если уменьшить ширину окна браузера, один или несколько элементов `<div>` могут “перепрыгнуть” на строку ниже.

Это происходит из-за того, что элементы `<div>` не содержат иного контейнера, кроме окна браузера. Структура подобного типа называется *текучей* (liquid layout).

Тем не менее элементам `<div>` можно запретить перепрыгивать на другие строки. Если поместить их в контейнер фиксированной ширины, они будут оставаться на месте; при этом все равно, насколько узким станет окно браузера. Это *структура с фиксированной шириной*.

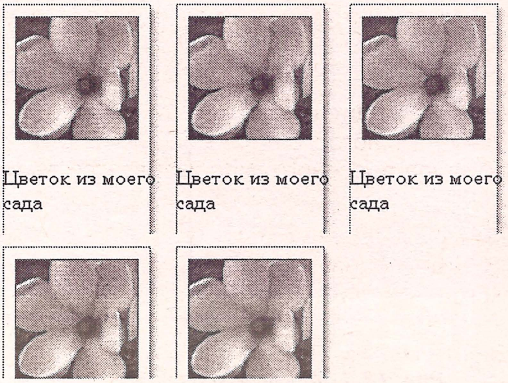
Стилевое оформление изображения

Сейчас изображения жестко связаны с границами своих контейнеров — элементов `<div>`. Чтобы дать им определенную свободу, можно установить поля сверху и слева, используя свойство `margin: 10px 0 0 10px`.

Кроме того, можно добавить границу к изображению, используя свойство `border: 1px solid #777`, как показано в листинге 12.4. Результаты выполнения данного кода представлены на рис. 12.3.

Листинг 12.4. Код CSS со стилевым оформлением изображения

```
div.thumbnail
{
    width: 130px;
    float: left;
    margin: 0 10px 10px 0;
    background: url(chapter12a.gif) no-repeat;
}
div.thumbnail img
{
    margin: 10px 0 0 10px;
    border: 1px solid #777;
}
```



Цветок из моего сада

Цветок из моего сада

Цветок из моего сада

Рис. 12.3. Изображения с переопределенным стилем

Стилевое оформление элемента абзаца

Большинство браузеров представляет стандартные элементы <p> с полями `1em` сверху и снизу. Чтобы убрать эти поля, используйте свойство `margin: 0`.

Сейчас абзац располагается под изображением, но все еще вплотную к границе контейнера. Чтобы добавить пустое пространство, следует использовать заполнение. В данном случае мы применяем 20 пикселей заполнения справа, 30 пикселей заполнения снизу и 10 пикселей — слева.

Далее необходимо добавить фоновое изображение, чтобы завершить иллюзию рамки, отбрасывающей тень. Данное изображение, показанное на рис. 12.4, будет располагаться над элементом <p>.

Для изображения необходимо задать свойство `no-repeat`, чтобы оно не повторялось в длинной подписи. Вертикальное положение фона устанавливается равным `100%`, в результате чего нижняя граница изображения будет выровнена по нижней границе элемента `<p>`. Соответствующий код показан в листинге 12.5, результат его выполнения иллюстрируется на рис. 12.5.

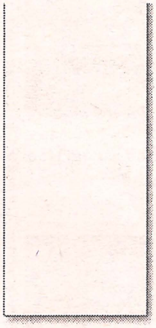


Рис. 12.4. Фоновое изображение, используемое элементом <p>

Листинг 12.5. Код CSS, определяющий стилевое оформление элемента <p>

```
div.thumbnail
{
    width: 130px;
    float: left;
    margin: 0 10px 10px 0;
    background: url(chapter12a.gif) no-repeat;
}
div.thumbnail img
{
    border: 1px solid #777;
    margin: 10px 0 0 10px;
}
div.thumbnail p
{
    margin: 0;
    padding: 0 20px 30px 10px;
    background: url(chapter12b.gif) no-repeat 0 100%;
}
```

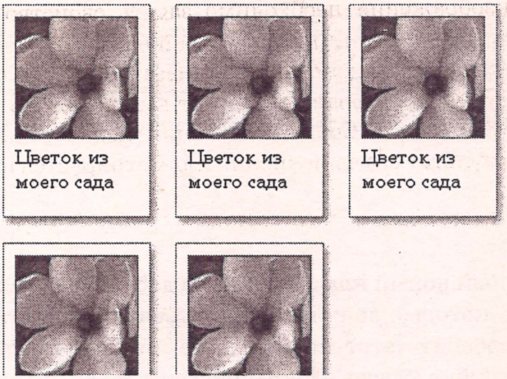


Рис. 12.5. Окончательный вид галереи эскизов

Принудительный перевод строки

Иногда требуется разместить несколько эскизов в одной строке. Для этого необходимо создать новый класс, а затем применить его к конкретным элементам `<div>`. Данный новый класс будет иметь одно объявление — `clear: left`. Эта команда переведет элемент `<div>` вниз на новую строку, расположенную ниже нижней границы всех предшествующих элементов `<div>`. Соответствующий код CSS показан в листинге 12.6.

Листинг 12.6. Код CSS с принудительным переводом строки

```
div.thumbnail
{
    width: 130px;
    float: left;
    margin: 0 10px 10px 0;
    background: url(chapter12a.gif) no-repeat;
}
div.thumbnail img
{
    border: 1px solid #777;
    margin: 10px 0 0 10px;
}
div.thumbnail p
{
    margin: 0;
    padding: 0 20px 30px 10px;
    background: url(chapter12b.gif) no-repeat 0 100%;
}
.clear
{
    clear: left;
}
```

Описанный новый класс требуется добавить ко всем элементам `<div>`, которые должны начинаться с новой строки. Проще всего добавить этот новый класс, включив его в существующий атрибут класса. Таким образом, `class="thumbnail"` можно заменить классом `class="thumbnail clear"`, как показано в листинге 12.7. Окончательный результат представлен на рис. 12.6.

Листинг 12.7. Код HTML с новым классом

```
<div class="thumbnail">
    <img src="chapter12c.jpg" alt="">
    <p>A flower from my garden</p>
</div>
<div class="thumbnail">
    <img src="chapter12c.jpg" alt="">
    <p>A flower from my garden</p>
</div>
<div class="thumbnail">
    <img src="chapter12c.jpg" alt="">
    <p>A flower from my garden</p>
</div>
<div class="thumbnail clear">
    <img src="chapter12c.jpg" alt="">
    <p>A flower from my garden</p>
</div>
<div class="thumbnail">
    <img src="chapter12c.jpg" alt="">
    <p>A flower from my garden</p>
</div>
```

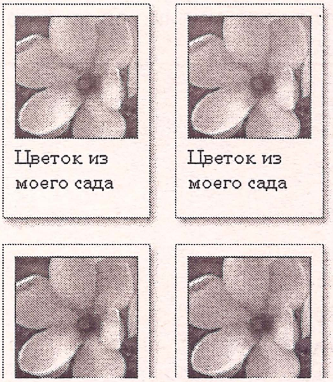


Рис. 12.6. Галерея эскизов

Размещение подписей рядом

Используя те же селекторы и код HTML, можно изменить структуру страницы так, чтобы изображения и их подписи располагались рядом.

Чтобы реализовать эту идею, вначале необходимо увеличить ширину элемента `<div>`, чтобы он, помимо изображения, вмещал подпись. Для этого соответствующее объявление заменяется объявлением `width: 250px`.

Далее к `<div>` добавляются нижнее заполнение (свойство `padding-bottom`) и граница (`border`).

Объявление `background` можно убрать вообще.

Изображение должно “всплывать” влево, чтобы подпись могла располагаться рядом с ним. Ширину в данном случае определять не нужно, поскольку изображение имеет собственную ширину.

Наконец изображению требуется небольшое поле, чтобы подпись не перекрывала его. Как показано в листинге 12.8, для всех сторон, кроме нижней, величину поля можно выбрать равной `10px`. Результаты выполнения данного кода представлены на рис. 12.7.

Листинг 12.8. Код CSS для размещения подписи рядом с рисунком

```
div.thumbnail
{
    float: left;
    width: 250px;
    margin: 0 10px 10px 0;
    padding-bottom: 10px;
    border: 1px solid #777;
}
div.thumbnail img
{
    float: left;
    border: 1px solid #777;
    margin: 10px 10px 0 10px;
}
div.thumbnail p
{
    margin: 0;
    padding: 10px;
}
```

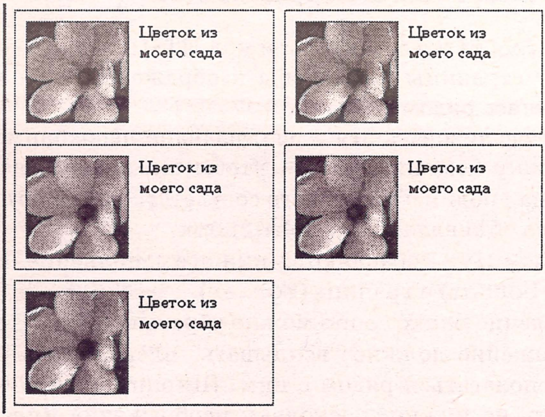
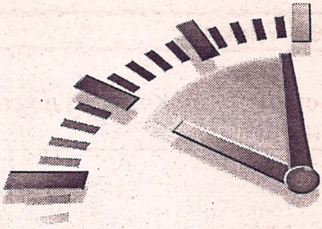

Цветок из моего сада
Цветок из моего сада
Цветок из моего сада
Цветок из моего сада
Цветок из моего сада

Рис. 12.7. Размещение подписи рядом с рисунком

Резюме

Из этого урока вы узнали, как сделать “плавающими” ряд элементов `<div>` и определить для них стиль отображения. Кроме того, вы научились принудительно начинать новую строку с помощью `clear`. В следующем уроке вы научитесь определять стиль цитаты.

Урок 13



Стилевое оформление блока цитаты

В этом уроке вы научитесь определять стиль блока цитаты, используя два фоновых изображения. Наша цель — создать цитату внутри двух больших (графических) кавычек.

Применение элемента `<blockquote>`

Элемент `<blockquote>` часто используется для создания текста с отступом. Это недопустимо! Его следует применять только для разметки длинных цитат, состоящих из элементов блочного уровня.

В данном уроке требуются цитата и источник или автор цитаты. Данные две строки можно пометить как абзацы, а затем поместить внутрь элемента `<blockquote>`. Абзацу, содержащему информацию об источнике, будет присвоен класс `source`. В данном случае используется именно класс, а не идентификатор, поскольку может потребоваться визуально выделить на одной странице несколько элементов `<blockquote>`.

Более подробную информацию об источнике (например, Web-адрес) можно добавить в элемент `<blockquote>`, используя атрибут `cite`, как показано в листинге 13.1.

Листинг 13.1. Код HTML, содержащий разметку для иллюстрации цитаты

```
<blockquote cite="http://www.sitename.com/quote/smith.htm">
    <p>
        Lorem ipsum dolor ...
    </p>
    <p class="source">
        John Smith
    </p>
</blockquote>
```

Семантически правильная разметка и блоки цитат

Семантически правильная разметка связана с пониманием HTML-элементов и их значения. Также она относится к использованию этих элементов для придания смысла контексту, который в них содержится.

Если вы используете семантически правильную разметку, ваше содержимое будет иметь смысл на множестве устройств, включая текстовые браузеры, программы для чтения с экрана и ручные устройства.

Если вы используете семантически плохую разметку, ваше содержимое либо будет бессмысленным, либо будет неправильно передаваться на множестве устройств.

Примером семантически плохой разметки является использование элемента `<blockquote>` для создания текста с отступом. В таком случае элемент `<blockquote>` не добавит смысла, а изменит представление содержимого. Даже хуже: в содержимое с отступом будет вложен неправильный смысл.

Гораздо лучше было бы сформировать отступ, используя CSS, и сохранить `<blockquote>` для первоначальной цели — форматирования длинных цитат.

Чтобы определить стиль элемента `<blockquote>` и его содержимого, используются три селектора.

- Селектор типа, используемый для отбора всех экземпляров `<blockquote>` на данной странице.
- Селектор-потомок, используемый для отбора всех элементов `<p>`, располагающихся внутри `<blockquote>`.

- Еще один селектор-потомок, используемый для отбора всех элементов `<p>`, стиль которых определен с помощью класса `source`.

Данные селекторы показаны в листинге 13.2.

Листинг 13.2. Код CSS, иллюстрирующий селекторы, которые используются в стилевом оформлении элемента `<blockquote>`

```
blockquote {...}
blockquote p {...}
blockquote p.source {...}
```

Стилевое оформление элемента <blockquote>

По умолчанию элементы `<blockquote>` представляются с отступами слева и справа. Такое положение можно изменить, переустановив поля. В данном случае верхнее и нижнее поля устанавливаются равными `1em`, а левое и правое — равными `0`. Для реализации этого можно использовать сокращенную форму объявления полей — `margin: 1em 0`.

Далее необходимо применить границу к элементу `<blockquote>`, чтобы отделить его от другого содержимого страницы. Используем светло-серую границу шириной 1 пиксель: `border: 1px solid #ddd`.

Первое фоновое изображение будет применяться непосредственно к элементу `<blockquote>`. Соответствующее объявление выглядит следующим образом: `background-image: url(lesson13.gif)`.

Выбранное изображение будет размещено в левом верхнем углу элемента `<blockquote>`. Для этого координатам по осям *x* и *y* присваивается значение 5 пикселей с помощью объявления `background-position: 5px 5px`.

Чтобы запретить изображению повторяться по всему элементу `<blockquote>`, добавьте свойство `background-repeat: no-repeat`.

Указанные три объявления свойств фона можно объединить в одно, используя `background: url(lesson13.gif)`

`5px 5px no-repeat`. Получающееся в результате фоновое изображение показано на рис. 13.1.

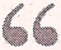
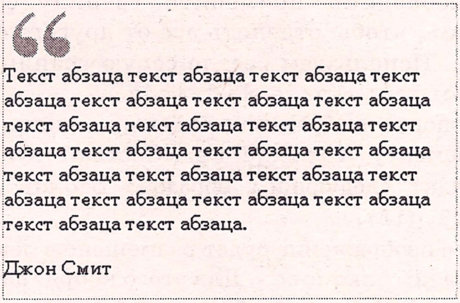


Рис. 13.1. Фоновое изображение, соотнесенное с элементом `<blockquote>`

Теперь изображение находится на месте, но текст располагается поверх него. Чтобы это исправить, необходимо добавить заполнение вверху, `<blockquote>`, используя `padding-top: 30px`, как показано в листинге 13.3. Результаты выполнения данного кода представлены на рис. 13.2.

Листинг 13.3. Код CSS со стилевым оформлением элемента `<blockquote>`

```
blockquote
{
    margin: 1em 0;
    border: 1px solid #ddd;
    background: url(lesson13.gif) 5px 5px no-repeat;
    padding-top: 30px;
}
```



Текст абзаца текст абзаца.

Джон Смит

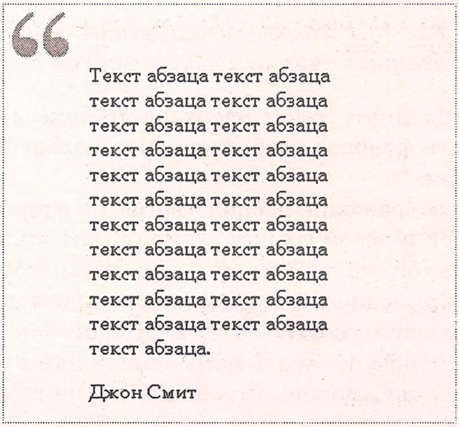
Рис. 13.2. Элемент `<blockquote>` с переопределенным стилем

Стилевое оформление абзаца

Далее необходимо добавить заполнение к левой и правой границам всех абзацев, принадлежащих элементу `<blockquote>`. Таким образом, содержимое отодвинется от границ. Используем свойство `padding: 0 70px`, как показано в листинге 13.4. Результаты выполнения данного кода представлены на рис. 13.3.

Листинг 13.4. Код CSS со стилевым оформлением абзаца

```
blockquote
{
    margin: 1em 0;
    border: 1px solid #ddd;
    background: url(lesson13.gif) 5px 5px no-repeat;
    padding-top: 30px;
}
    blockquote p
{
    padding: 0 70px;
}
```



Текст абзаца текст абзаца
текст абзаца текст абзаца
текст абзаца текст абзаца
текст абзаца текст абзаца
текст абзаца текст абзаца
текст абзаца текст абзаца
текст абзаца текст абзаца
текст абзаца текст абзаца
текст абзаца текст абзаца
текст абзаца текст абзаца
текст абзаца текст абзаца
текст абзаца.

Джон Смит

Рис. 13.3. Абзац с переопределенным стилем

Стилевое оформление класса `source`

Теперь стили элемента `<blockquote>` и абзацев определены, и можно сосредоточиться на абзаце класса `source`. С помощью данного абзаца мы поместим в правом нижнем углу цитаты второе фоновое изображение.

Чтобы задать изображение, режим повтора и положение изображения, можно использовать сокращенную форму записи: `background: url(lesson13a.gif) no-repeat 100% 100%`. Положения по осям x и y должны устанавливаться равными `100%`, чтобы правый нижний угол изображения находился в правом нижнем углу абзаца. Чтобы изображение не повторялось под текстом, задается режим `no-repeat`. Использованное нами фоновое изображение показано на рис. 13.4.

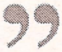


Рис. 13.4. Фоновое изображение, соотнесенное с абзацем класса `source`

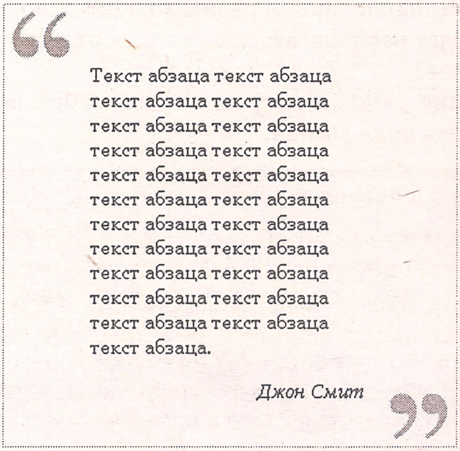
Чтобы отодвинуть текст вверх на 30 пикселей и не дать ему перекрыть фоновое изображение, используйте `padding-bottom: 30px`.

Верхнее изображение располагается на 5 пикселей к центру от верхней и левой границ элемента `<blockquote>`. Чтобы создать такой же эффект для нижнего изображения, добавьте поле 5px к правой и нижней границам абзаца класса `source`, используя `margin: 0 5px 5px 0`.

Наконец, чтобы отличать измененный нами текст от других цитат в блоке, можно выровнять его по правой границе и представить курсивом. Для этого используются свойства `text-align: right` и `font-style: italic`, как показано в листинге 13.5. Результаты выполнения данного кода представлены на рис. 13.5.

Листинг 13.5. Код CSS со стилевым оформлением абзаца класса `source`

```
blockquote
{
    margin: 1em 0;
    border: 1px solid #ddd;
    background: url(lesson13.gif) 5px 5px no-repeat;
    padding-top: 30px;
}
    blockquote p
{
    padding: 0 70px;
}
blockquote p.source
{
    background: url(lesson13a.gif) no-repeat 100% 100%;
    padding-bottom: 30px;
    margin: 0 5px 5px 0;
    text-align: right;
    font-style: italic;
}
```



Текст абзаца текст абзаца.

Джон Смит

Рис. 13.5. Абзац класса `source` *с переопределенным стилем*

Создание другого варианта

С помощью тех же селекторов и кода HTML можно изменить стиль так, чтобы элемент `<blockquote>` и его содержимое выглядели совершенно иначе.

Для примера создадим элемент `<blockquote>` фиксированной ширины с одним большим фоновым изображением, как показано на рис 13.6.

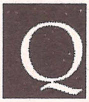


Рис. 13.6. Новое фоновое изображение

В селекторе `blockquote` необходимо изменить фоновое изображение и его положение. В данном случае положения по осям *x* и *y* устанавливаются равными `0` (это означает, что изображение будет располагаться в левом верхнем углу элемента `<blockquote>`).

Объявление `padding-top` меняется с `30px` на `1px`, чтобы зафиксировать поля абзаца.

Поля в ловушке

Стандартный абзац имеет предопределенные верхнее и нижнее поля.

Когда абзац помещается в другой контейнер, верхнее поле абзаца может стать источником проблем, поскольку одни браузеры отображают абзац и верхнее поле внутри контейнера, а другие отображают только абзац, позволяя полю пересекать верхнюю границу контейнера.

Чтобы изменить такое поведение, можно применить к контейнеру свойство `border-top` или `padding-top` со значением `1px`.

Данное действие называется фиксацией полей (trapping margins).

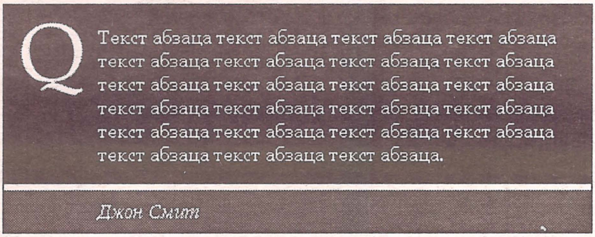
Селектор `blockquote.p` меняется с `padding: 0 70px` на `padding: 0 1em 0 80px`. Данное новое заполнение отодвинет текст от левой границы элемента `<blockquote>`, освободив место для нового фонового изображения.

Селектор `blockquote p.source` меняется еще больше. Фоновое изображение удаляется совсем. Поля устанавливаются равными `0`. К верхней границе абзаца добавляется белая граница шириной `5px`, отделяющая источник от цитаты. Заполнение меняется с `padding-bottom: 30px` на `padding: .5em .5em .5em 80px`.

Наконец свойство `text-align: right` удаляется и добавляется свойство `background: #336`. Результаты изменений показаны в листинге 13.6. Результаты выполнения данного кода представлены на рис. 13.7.

Листинг 13.6. Код CSS для другого варианта стиля `<blockquote>`

```
blockquote
{
    margin: 1em 0;
    border: 1px solid #000;
    background: #000 url(lesson13b.gif) no-repeat 0 0;
    padding-top: 1px;
    color: #fff;
    width: 500px;
}
blockquote p
{
    padding: 0 1em 0 80px;
}
blockquote p.source
{
    margin: 0;
    border-top: 5px solid #fff;
    padding: .5em .5em .5em 80px;
    background: #336;
    font-style: italic;
}
```



Текст абзаца текст абзаца.

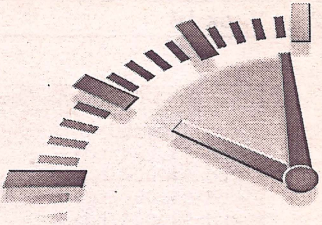
Джон Смит

Рис. 13.7. Один из вариантов оформления стиля `<blockquote>`

Резюме

Из этого урока вы узнали, как поместить длинную цитату и ее источник в элемент `<blockquote>`, а затем оформить их различными стилями. В следующем уроке вы научитесь размечать таблицы и определять для них стили.

Урок 14



Стилевое оформление таблицы

Данный урок разбит на две части. Вначале вы научитесь добавлять к данным элементы, повышающие их доступность для программ чтения с экрана, а затем — определять стили различных элементов таблицы.

Стартовая таблица

По мере роста популярности CSS наблюдается все более сильная тенденция к отказу от таблиц как средства разметки контекста. Тем не менее было время, когда таблицы представляли собой лучшее средство разметки, особенно для данных, визуализируемых в форме матрицы. В данном уроке мы будем использовать для иллюстрации таблицу, представленную в листинге 14.1.

Листинг 14.1. Код HTML, содержащий разметку таблицы данных

```
<table>
    <tr>
        <td>Item</td>
        <td>Threaded screws </td>
        <td>Flat nails</td>
        <td>Dyna-bolts </td>
        <td>Spring washers</td>
    </tr>
    <tr>
        <td>1 kg</td>
        <td>$2.50</td>
        <td>$3.50</td>
        <td>$4.50</td>
        <td>$2.50</td>
    </tr>
    <tr>
        <td>2kg</td>
        <td>$3.00</td>
        <td>$4.00</td>
        <td>$5.00</td>
        <td>$3.00</td>
    </tr>
    <tr>
        <td>3kg</td>
        <td>$3.50</td>
        <td>$4.50</td>
        <td>$5.50</td>
        <td>$3.50</td>
    </tr>
    <tr>
        <td>4kg</td>
        <td>$4.00</td>
        <td>$5.00</td>
        <td>$6.00</td>
        <td>$4.00</td>
    </tr>
</table>
```

Добавление к таблице элементов, повышающих ее доступность

К таблице можно добавить несколько элементов, которые сделают ее более доступной.

В сложных таблицах следует использовать атрибут `summary`, показанный в листинге 14.2, поскольку он предлагает четкое описание того, что представляет таблица. Он не отображается на экране современных (созданных согласно существующим стандартам) Web-браузеров, но его можно отображать на других устройствах навигации в Web — ручных устройствах, сотовых телефонах и т.д. Чтобы сделать код доступным для пользователей, применяющих невизуальные устройства, используется атрибут `summary`.

Листинг 14.2. Код HTML, демонстрирующий использование атрибута `summary`

```
<table summary="Table of screws, Flat nails, Dyna-bolts and
Spring washers, in kilos">
```

В любой таблице данных должен присутствовать элемент `caption`, кратко описывающий содержимое таблицы. В отличие от `summary`, элемент `caption` отображается на экране (обычно отцентрированным над таблицей). Как показано в листинге 14.3, данный элемент должен располагаться непосредственно после тега, открывающего таблицу.

Листинг 14.3. Код HTML, демонстрирующий использование элемента `caption`

```
<caption>
    Pricing for threaded screws, flat nails, dyna-bolts and

spring washers
</caption>
```

Элемент `<th>`, показанный в листинге 14.4, должен использоваться для определения строки или столбца с заголовком в таблице данных. Таким образом, создаются связи между элементами `<th>` и `<td>`, что очень важно для невизуальных устройств.

Листинг 14.4. Код HTML, демонстрирующий использование элементов `<th>`

```
<th>Item</th>
<th>Threaded screws </th>
<th>Flat nails</th>
<th>Dyna-bolts </th>
<th>Spring washers</th>
```

Элементы `<thead>`, `<tbody>` и `<tfoot>`, показанные в листинге 14.5, используются для группировки строк в таблицы. Элементы `<thead>` и `<tfoot>` должны содержать информацию о столбцах таблицы, а `<tbody>` — собственно данные таблицы.

Листинг 14.5. Код HTML, демонстрирующий использование элементов `<thead>` и `<tbody>`

```
<thead>
    <tr>
        <th>Item</th>
        <th>Threaded screws </th>
        <th>Flat nails</th>
        <th>Dyna-bolts </th>
        <th>Spring washers</th>
    </tr>
</thead>
<tbody>
    <tr>
        <th>1 kg</th>
        <td>$2.50</td>
        <td>$3.50</td>
        <td>$4.50</td>
        <td>$2.50</td>
    </tr>
</tbody>
```

Атрибут `abbr`, показанный в листинге 14.6, используется для создания сокращенной формы содержимого важных ячеек. Атрибут `abbr` важен для людей, использующих программы чтения с экрана, которым может потребоваться, чтобы содержимое ячейки громко считывалось несколько раз подряд.

Листинг 14.6. Код HTML, иллюстрирующий использование атрибутов `abbr`

```
<tr>
    <th>Item</th>
    <th abbr="screws">Threaded screws</th>
    <th abbr="nails">Flat nails</th>
    <th abbr="bolts">Dyna-bolts</th>
    <th abbr="washers">Spring washers</th>
</tr>
```

С помощью атрибутов `headers` и `id` ячейки таблицы связываются с соответствующим заголовком. Каждому заголовку должен присваиваться уникальный идентификатор `id`. После этого, как показано в листинге 14.7, к каждому элементу `<td>` добавляется атрибут `headers`.

Листинг 14.7. Код HTML, демонстрирующий использование атрибутов `headers` и `id`

```
<table summary="Table of screws, Flat nails, Dyna-bolts and

Spring washers, in kilos">
    <caption>
        Pricing for threaded screws, flat nails, dyna-bolts
and spring washers
    </caption>
    <thead>
        <tr>
```

```
          <th>Item</th>
          <th id="screws" abbr="screws">Threaded
screws</th>
          <th id="nails" abbr="nails">Flat nails</th>
          <th id="bolts" abbr="bolts">Dyna-bolts</th>
          <th id="washers" abbr="washers">Spring
          washers</th>
        </tr>
    </thead>
    <tbody>
        <tr>
          <th id="one">1 kg</th>
          <td headers="screws one">$2.50</td>
          <td headers="nails one">$3.50</td>
          <td headers="bolts one">$4.50</td>
          <td headers="washers one">$2.50</td>
        </tr>
        <tr>
          <th id="two">2kg</th>
          <td headers="screws two">$3.00</td>
          <td headers="nails two">$4.00</td>
          <td headers="bolts two">$5.00</td>
          <td headers="washers two">$3.00</td>
        </tr>
        <tr>
          <th id="three">3kg</th>
          <td headers="screws three">$3.50</td>
          <td headers="nails three">$4.50</td>
          <td headers="bolts three">$5.50</td>
          <td headers="washers three">$3.50</td>
        </tr>
        <tr>
          <th id="four">4kg</th>
          <td headers="screws four">$4.00</td>
          <td headers="nails four">$5.00</td>
          <td headers="bolts four">$6.00</td>
          <td headers="washers four">$4.00</td>
        </tr>
    </tbody>
</table>
```

Создание селекторов для формирования стилевого оформления таблицы

Чтобы определить стиль таблицы и ее содержимого, используются восемь селекторов, как показано в листинге 14.8.

Листинг 14.8. Код CSS, иллюстрирующий селекторы, которые используются для создания стиля таблицы

```
caption {...}
table {...}
th, td {...}
tr {...}
thead th {...}
tbody th {...}
tr.alternate {...}
tr.alternate th {...}
```

Стилевое оформление подписи

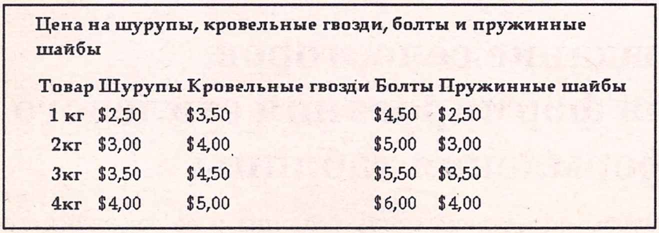
По умолчанию подпись таблицы отображается над таблицей. В большинстве современных браузеров подпись выравнивается по центру, но это положение можно изменить, используя, например, свойство `text-align: left`.

Чтобы увеличить отступ между подписью и таблицей, свойству `margin-bottom` присваивается значение `.5em`.

Кроме того, подпись можно сделать немного крупнее, чтобы она отличалась от шрифта содержимого таблицы. Для этого используется свойство `font-weight:bold`, как показано в листинге 14.9. Результаты выполнения данного кода представлены на рис. 14.1.

Листинг 14.9. Код CSS, определяющий стилевое оформление подписи

```
caption
{
    text-align: left;
    margin: 0 0 .5em 0;
    font-weight: bold;
}
```



Цена на шурупы, кровельные гвозди, болты и пружинные шайбы

Товар	Шурупы	Кровельные гвозди	Болты	Пружинные шайбы
1 кг	$2,50	$3,50	$4,50	$2,50
2кг	$3,00	$4,00	$5,00	$3,00
3кг	$3,50	$4,50	$5,50	$3,50
4кг	$4,00	$5,00	$6,00	$4,00

Рис. 14.1. Подпись таблицы с измененным стилем

Стилевое оформление элемента `<table>`

Чтобы убрать отступы между ячейками, к элементу `<table>` применяется свойство `border-collapse: collapse`, как показано в листинге 14.10.

Листинг 14.10. Код CSS, определяющий стилевое оформление элемента `<table>`

```
caption
{
    text-align: left;
    margin: 0 0 .5em 0;
    font-weight: bold;
}
table
{
    border-collapse: collapse;
}
```

Таблицы и расстояние между ячейками

Стандартная таблица имеет промежуток между ячейками, равный примерно `2px`. Удалить его можно двумя способами.

Первый способ заключается в применении `cellspacing="0"` как атрибута элемента `<table>`. Это решение неидеально, поскольку к таблице добавляется атрибут представления. Если вы захотите позже изменить стиль таблицы, то вам придется изменять и HTML, и CSS.

Второй способ состоит в применении к элементу `<table>` свойства `border-collapse: collapse`. Данный метод предпочтительнее, поскольку используются только средства CSS и внешний вид таблицы можно изменить, не затрагивая HTML-код.

Стилевое оформление элементов `<th>` и `<td>`

Теперь в стиль элементов `<th>` и `<td>` необходимо включить правую и левую границы. Поскольку граница будет появляться во всех элементах `<td>` и `<th>`, эти элементы можно сгруппировать в один селектор.

Использование границы лучше использования атрибута `cellspacing`, поскольку в любой момент можно, например, изменить ее цвет или ширину.

Чтобы применить заполнение ко всем ячейкам, используйте свойство `padding: .5em`, как показано в листинге 14.11.

Листинг 14.11. Код CSS, определяющий стилевое оформление элементов `<th>` и `<td>`

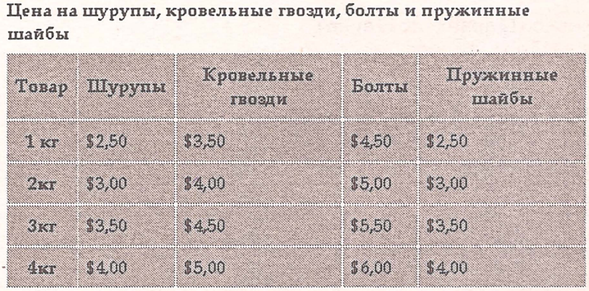
```
caption
{
    text-align: left;
    margin: 0 0 .5em 0;
    font-weight: bold;
}
table
{
    border-collapse: collapse;
}
th, td
{
    border-right: 1px solid #fff;
    border-bottom: 1px solid #fff;
    padding: .5em;
}
```

Стилевое оформление элемента `<tr>`

Как показано в листинге 14.12, стиль элемента `<tr>` удобно определить с использованием свойства `background-color`. Если выбранный нами цвет вам не нравится, вы можете его изменить. Результаты выполнения данного кода представлены на рис. 14.2.

Листинг 14.12. Код CSS, определяющий стилевое оформление элемента `<tr>`

```
caption
{
    text-align: left;
    margin: 0 0 .5em 0;
    font-weight: bold;
}
table
{
        border-collapse: collapse;}
th, td
{
    border-right: 1px solid #fff;
    border-bottom: 1px solid #fff;
    padding: .5em;
}
tr
{
    background: #B0C4D7;
}
```



Цена на шурупы, кровельные гвозди, болты и пружинные шайбы

Товар	Шурупы	Кровельные гвозди	Болты	Пружинные шайбы
1 кг	$2,50	$3,50	$4,50	$2,50
2кг	$3,00	$4,00	$5,00	$3,00
3кг	$3,50	$4,50	$5,50	$3,50
4кг	$4,00	$5,00	$6,00	$4,00

Рис. 14.2. Элемент `<tr>` с переопределенным стилем

Выбор экземпляров элемента `<th>`

Следующий шаг заключается в создании фоновых цветов для элемента `<th>`. Используя селекторы-потомки, элементы `<th>` вверху и слева от таблицы можно окрасить другим цветом.

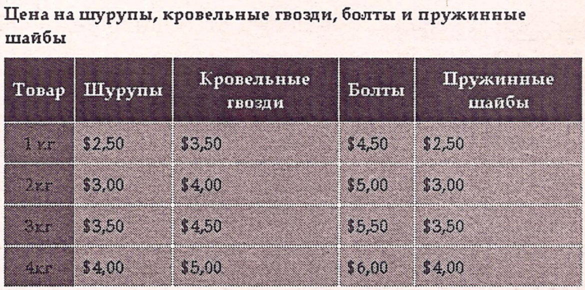
Стиль элементов `<th>` в верхней строке задается с использованием `thead th {...}`, поскольку они находятся внутри элемента `<thead>`.

Стиль элементов `<th>` в левом столбце определяется с использованием `tbody th {...}`, поскольку они находятся внутри элемента `<tbody>`.

Кроме того, как показано в листинге 14.13, для элементов `<th>` левого столбца можно задать свойство `font-weight: normal`, чтобы отличать их от заголовка, представленного в верхней строке. Результаты выполнения данного кода представлены на рис. 14.3.

Листинг 14.13. Код CSS, определяющий стилевое оформление элементов `<th>`

```
caption
{
    text-align: left;
    margin: 0 0 .5em 0;
    font-weight: bold;
}
table
{
    border-collapse: collapse;
}
th, td
{
    border-right: 1px solid #fff;
    border-bottom: 1px solid #fff;
    padding: .5em;
}
tr
{
    background: #B0C4D7;
}
thead th
{
    background: #036;
    color: #fff;
}
tbody th
{
    font-weight: normal;
    background: #658CB1;
}
```



Цена на шурупы, кровельные гвозди, болты и пружинные шайбы

Товар	Шурупы	Кровельные гвозди	Болты	Пружинные шайбы
1кг	$2,50	$3,50	$4,50	$2,50
2кг	$3,00	$4,00	$5,00	$3,00
3кг	$3,50	$4,50	$5,50	$3,50
4кг	$4,00	$5,00	$6,00	$4,00

Рис. 14.3. *Элементы `<th>` с переопределенными стилями*

Чередование цветов в строках

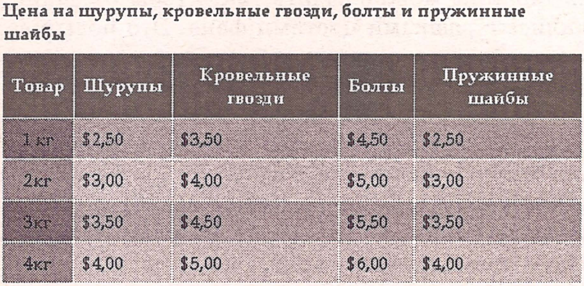
Стиль таблицы можно задать так, чтобы соседние строки отображались с разными цветами фона. Это повышает читабельность, особенно в длинных таблицах.

Для реализации этой идеи можно добавить класс к каждому второму элементу `<tr>`. В данном случае используем класс `alternate`. Элементы `<td>` и `<th>` в соседних строках получат несколько отличающиеся цвета фона. Как показано в листинге 14.14, требуемые селекторы имеют вид `tr.alternate td {...}` и `tr.alternate th {..}`. Результаты выполнения данного кода представлены на рис. 14.4.

Листинг 14.14. Код CSS, задающий стиль чередующихся строк

```
caption
{
    text-align: left;
    margin: 0 0 .5em 0;
    font-weight: bold;
}
table
{
    border-collapse: collapse;
}
th, td
{
    border-right: 1px solid #fff;
    border-bottom: 1px solid #fff;
    padding: .5em;
}
tr
```

```
{
    background: #B0C4D7;
}
thead th
{
    background: #036;
    color: #fff;
}
tbody th
{
    font-weight: normal;
    background: #658CB1;
}
tr.alternate
{
    background: #D7E0EA;
}
tr.alternate th
{
    background: #8AA9C7;
}
```



Цена на шурупы, кровельные гвозди, болты и пружинные шайбы

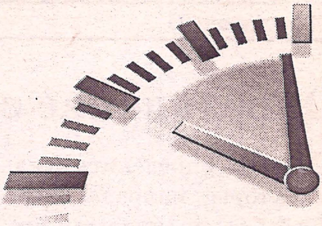
Товар	Шурупы	Кровельные гвозди	Болты	Пружинные шайбы
1кг	$2,50	$3,50	$4,50	$2,50
2кг	$3,00	$4,00	$5,00	$3,00
3кг	$3,50	$4,50	$5,50	$3,50
4кг	$4,00	$5,00	$6,00	$4,00

Рис. 14.4. В стиль строк добавлено чередование цветов

Резюме

Из этого урока вы узнали, как добавить к таблице элементы, повышающие доступность: `summary`, `caption`, `<thead>`, `<th>`, `headers` и `id`. Кроме того, вы научились изменять стиль таблицы и ее элементов. Данные элементы доступности не только улучшают вид Web-сайта на различных устройствах, но и позволяют использовать на этих устройствах CSS. В следующем уроке вы научитесь создавать вертикальную панель навигации.

Урок 15



Создание вертикальной панели навигации

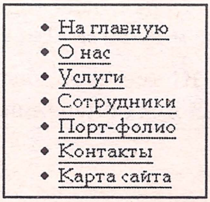
В этом уроке вы научитесь создавать вертикальную панель навигации. Кроме того, вы узнаете, как применять фоновые изображения и эффекты визуального выделения.

Зачем нужен список

По своей сути панель навигации сайта представляет просто набор ссылок на другие его страницы. Таким образом, для начала работы идеально подходит стандартный список HTML, описанный в листинге 15.1 и визуализированный на рис. 15.1.

Листинг 15.1. Код HTML, содержащий разметку списка

```
<ul id="navigation">
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Services</a></li>
    <li><a href="#">Staff</a></li>
    <li><a href="#">Portfolio</a></li>
    <li><a href="#">Contact</a></li>
    <li><a href="#">Sitemap</a></li>
</ul>
```



- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты
- Карта сайта

Рис. 15.1. Список без стиля

Определение стиля списка

Чтобы определить стиль списка, необходимо использовать селекторы, направленные на элементы `<ul>`, `<li>` и `<a>`. Чтобы гарантировать, что вы не захватите все экземпляры этих элементов, с каждым селектором необходимо задействовать уникальный идентификатор, `navigation`. Используемые в этом уроке четыре селектора показаны в листинге 15.2.

Листинг 15.2. Код CSS, иллюстрирующий селекторы, которые используются для определения стиля списка

```
ul#navigation {...}
ul#navigation a {...}
ul#navigation a:hover {...}
ul#navigation li {...}
```

Что такое селекторы

Селекторы используются для выбора элементов на HTML-странице, благодаря чему можно переопределять стили этих элементов. Подробнее об этом речь идет в уроке 3, "Селекторы в действии".

Стилевое оформление элемента

Большинство браузеров отображает списки HTML с отступом слева. Чтобы задать величину этого отступа, в одних браузерах применяется заполнение (Firefox, Netscape и Safari), а в других — поля (Internet Explorer и Opera). Чтобы убрать данный отступ во всех браузерах, необходимо обнулить значения свойств `padding-left` и `margin-left` в элементе `<ul>`, как показано в листинге 15.3.

Листинг 15.3. Код CSS для обнуления полей и заполнения

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
}
```

Чтобы убрать маркеры списка, присвойте свойству `list-style-type` значение `none`, как показано в листинге 15.4. Результаты выполнения данных правил CSS показаны на рис 15.2.

Листинг 15.4. Код CSS для удаления маркеров списка

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
```

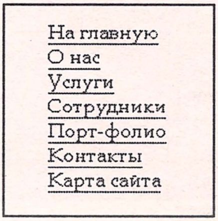


Рис. 15.2. Список с переопределенным элементом `<ul>`

Стилевое оформление элемента <a>

Текстовые ссылки обычно активны только при установлении указателя мыши на область текста. Чтобы увеличить эту активную область, примените к элементу `<a>` свойство `display: block;`. Это переведет элемент из категории внутритекстовых в разряд блочных и активная область займет всю ширину элемента списка.

Когда `<a>` является элементом блочного уровня, пользователям не обязательно щелкать на тексте; достаточно щелкнуть на любой области элемента списка. Присвоение элементам `<a>` свойства `display: block;` иллюстрируется в листинге 15.5.

Листинг 15.5. Код CSS для установки значения `display: block`

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
ul#navigation a
{
    display: block;
}
```

Чтобы убрать подчеркивание ссылок, используйте `text-decoration: none;` (листинг 15.6).

Листинг 15.6. Код CSS для удаления подчеркивания ссылок

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
ul#navigation a
{
    display: block;
    text-decoration: none;
}
```

Изменение поведения ссылок

Изменение стандартного поведения гиперссылок (например, удаление подчеркивания) может запутать пользователей, которые не распознают объект как ссылку.

Поэтому обычно удалять подчеркивание не рекомендуется, если вы не предлагаете пользователям какую-либо другую подсказку о том, что изображенный объект является ссылкой.

Чтобы установить цвет фона, можно использовать сокращенную форму записи соответствующего свойства: `background: #036;`, как показано в листинге 15.7. Если вам не нравится выбранный нами цвет, поменяйте его в любой момент.

Листинг 15.7. Код CSS для установки цвета фона

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
ul#navigation a
{
    display: block;
    text-decoration: none;
    background: #036;
}
```

Далее цвет текста устанавливается равным `#fff` (шестнадцатеричное представление белого цвета), как показано в листинге 15.8. Как и цвет фона, вы можете по своему усмотрению изменить цвет текста.

Листинг 15.8. Код CSS, устанавливающий цвет текста

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
ul#navigation a
{
    display: block;
    text-decoration: none;
    background: #036;
    color: #fff;
}
```

Далее потребуется заполнение `.2em` сверху и снизу от элемента `<a>` и заполнение `.5em` слева и справа. Чтобы не задавать эти величины в отдельных объявлениях, можно использовать сокращенную форму записи и задать их все сразу. В данном случае применяется свойство `padding: .2em .5em`, как показано в листинге 15.9.

Листинг 15.9. Код CSS, устанавливающий заполнение

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
ul#navigation a
{
    display: block;
    text-decoration: none;
    background: #036;
    color: #fff;
    padding: .2em .5em;
}
```

Зачем использовать единицы em

Заполнение, поля и ширину можно задавать в пикселях или единицах `em`. Второй вариант более гибкий, поскольку данные единицы масштабируются согласно выбранному пользователем размеру шрифта[1].

Чтобы создать пустое пространство между элементами списка, вверху каждого из них можно добавить границу. В данном случае используется свойство `border-bottom: #fff`, как показано в листинге 15.10.

Листинг 15.10. Код CSS, устанавливающий границы

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
ul#navigation a
{
    display: block;
    text-decoration: none;
    background: #036;
    color: #fff;
    padding: .2em .5em;
    border-bottom: 1px solid #fff;
}
```

Настройка border-bottom

В этом уроке свойству `border-bottom` присваивается значение `#fff` в предположении, что фон страницы белый. Цвет `border-bottom` необходимо выбирать таким же, как цвет страницы или фона контейнера.

Если требуется добавить больше места между элементами списка, ширину границы можно увеличить.

---

[1] *По определению 1 em — это ширина буквы “m” данного шрифта.* — Примеч. пер.

Для установки ширины элемента `<a>` используйте `width: 7em;`, как показано в листинге 15.11 и проиллюстрировано на рис. 15.3. Данную ширину вы можете изменить по собственному усмотрению.

Листинг 15.11. Код CSS для настройки ширины

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
ul#navigation a
{
    display: block;
    text-decoration: none;
    background: #036;
    color: #fff;
    padding: .2em .5em;
    border-bottom: 1px solid #fff;
    width: 7em;
}
```

Ширина списка

Как обсуждалось в уроке 5, "Блочная модель CSS", окончательная ширина блока получается после добавления заполнения к области содержимого. В данном случае заполнение `.5em` применяется слева и справа от списка, поэтому ширина элементов списка становится равной `8em` (ширина содержимого `7em` плюс два поля по `.5em`).

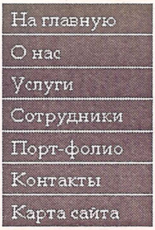


Рис. 15.3. Список с переопределенным стилем элемента <a>

Исправление странных границ

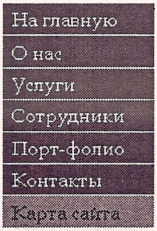
Определенные браузеры, например Netscape, Mozilla и Firefox, неправильно визуализируют свойство `border-bottom` для некоторых элементов списка — обычно для тех, которые находятся в середине списка. Чтобы исправить положение, можно толщину границы `1px` установить равной `1em`.

Добавление эффекта визуального выделения

Чтобы изменить стиль ссылок при их визуальном выделении, можно использовать псевдокласс `:hover`. В данном случае установим цвет фона равным `#69C`, а цвет теста — равным `#000`, как показано в листинге 15.12. Результаты выполнения данного кода представлены на рис. 15.4. Если вам не нравится такой подбор цветов, можете его изменить.

Листинг 15.12. Код CSS с настройкой визуального выделения

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
ul#navigation a
{
    display: block;
    text-decoration: none;
    background: #036;
    color: #fff;
    padding: .2em .5em;
    border-bottom: 1px solid #fff;
    width: 7em;
}
ul#navigation a:hover
{
    background: #69C;
    color: #000;
}
```



На главную
О нас
Услуги
Сотрудники
Порт-фолио
Контакты
Карта сайта

Рис. 15.4. Список с измененным визуальным выделением

Стилевое оформление элемента `<li>`

Возможно, вы заметили, что в некоторых версиях Internet Explorer для Windows или Opera наблюдаются небольшие промежутки между элементами списка. Это можно исправить, добавив к элементу `<li>` свойство `display: inline` (листинг 15.13).

Листинг 15.13. Код CSS для установки свойства `display: inline` для элемента `<li>`

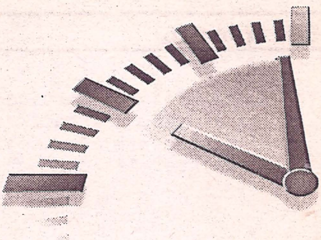
```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
ul#navigation a
{
    display: block;
    text-decoration: none;
    background: #036;
    color: #fff;
    padding: .2em .5em;
    border-bottom: 1px solid #fff;
    width: 7em;
}
ul#navigation a:hover
{
    background: #69C;
    color: #000;
}
```

```
ul#navigation li
{
    display: inline;
}
```

Резюме

Из этого урока вы узнали, что, изменяя стиль списков, можно создать вертикальную панель навигации, научились устанавливать объявления для элемента <a> и использовать псевдокласс `:hover`. В следующем уроке вы научитесь создавать горизонтальную панель навигации.

Урок 16



Создание горизонтальной панели навигации

В этом уроке вы научитесь создавать горизонтальную панель навигации из стандартного списка HTML. Кроме того, вы узнаете, как сделать “плавающими” элемент <ul> (для создания навигационной панели) и элемент <a> (для создания ряда квадратных кнопок с тонкими разделяющими линиями вдоль правой границы).

Стилевое оформление списка

Чтобы создать стиль списка, необходимо использовать селекторы, направленные на элементы `<ul>`, `<li>` и `<a>`. Кроме того, в каждый селектор необходимо включить уникальный идентификатор `navigation`. Используемые в этом уроке четыре селектора показаны в листинге 16.1. Код HTML, с которым мы будем работать, показан в листинге 16.2.

Листинг 16.1. Код CSS, иллюстрирующий селекторы, которые используются для стилевого оформления списка

```
ul#navigation {...}
ul#navigation li {...}
ul#navigation a {...}
ul#navigation a:hover {...}
```

Листинг 16.2. Код HTML, содержащий разметку списка

```
<ul id="navigation">
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Services</a></li>
    <li><a href="#">Staff</a></li>
    <li><a href="#">Portfolio</a></li>
    <li><a href="#">Contact</a></li>
</ul>
```

Стилевое оформление элемента `<ul>`

Как обсуждалось в уроке 15, “Создание вертикальной панели навигации”, большинство браузеров отображает списки HTML с отступом слева. Чтобы убрать этот отступ, в элементе `<ul>` значения свойств `padding-left` и `margin-left` устанавливаются равными 0, как показано в листинге 16.3.

Листинг 16.3. Код CSS, обнуляющий поля и заполнение

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
}
```

Чтобы убрать маркеры списка, установите значение свойства `list-style-type` равным `none`, как показано в листинге 16.4.

Листинг 16.4. Код CSS, удаляющий маркеры списка

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
}
```

Далее добавляем цвет фона `#036`, используя сокращенную форму записи, как показано в листинге 16.5. При желании выбранный цвет можно изменить.

Листинг 16.5. Код CSS, устанавливающий цвет фона

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
    background: #036;
}
```

Чтобы сделать элемент `<ul>` “плавающим”, используйте `float: left`. Кроме того, необходимо установить его ширину. В данном случае используем значение `100%`, поскольку требуется, чтобы список распространялся на всю ширину страницы. Соответствующий код показан в листинге 16.6, результат его выполнения иллюстрируется на рис. 16.1. На этом этапе текст практически нечитаемый. Не волнуйтесь, все это будет исправлено после переопределения стиля элемента `<a>`.

Зачем нужно “плавание” элементам `<ul>` и `<a>`

В этом уроке элементы `<ul>` и `<a>` должны быть “плавающими”.

Элемент `<a>` “плавает” потому, что элементы списка расположены на горизонтальной линии и перекрывают друг друга.

Элемент `<ul>` должен быть “плавающим” потому, что он оборачивается вокруг элемента `<a>`. В противном случае он бы не имел высоты и был невидимым.

Листинг 16.6. Код CSS, задающий свойства и ширину “плавающего” элемента

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
    background: #036;
    float: left;
    width: 100%;
}
```

Рис. 16.1. Вид списка с заданным стилем элемента `<ul>`

Стилевое оформление элемента `<li>`

Чтобы гарантировать, что элементы списка отображаются в одной строке, элемент `<li>` должен устанавливаться со свойством `display: inline`, как показано в листинге 16.7. Результаты выполнения данного кода представлены на рис. 16.2.

Листинг 16.7. Код CSS, задающий свойство `display: inline;`

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
    background: #036;
    float: left;
    width: 100%;
}
ul#navigation li
{
    display: inline;
}
```

Рис. 16.2. Список с переопределенным стилем элемента `<li>`

Стилевое оформление элемента `<a>`

Чтобы увеличить активную область текстовых ссылок, можно применить свойство `display: block;` к элементу `<a>`. Это переведет данный элемент из разряда внутритекстовых в разряд блочных и позволит применить заполнение ко всем сторонам элемента.

Задайте элемент `<a>` со свойством `display: block;`. Теперь ко всем его сторонам можно будет применять заполнение. Таким образом, элемент получит дополнительную ширину и высоту, что увеличит область, на которой можно щелкнуть.

После этого элемент <a> следует сделать “плавающим”, чтобы все позиции списка выстраивались рядом в одной строке (листинг 16.8).

Листинг 16.8. Код CSS, устанавливающий свойство `display: block;`

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
    background: #036;
    float: left;
    width: 100%;
}
ul#navigation li
{
    display: inline;
}
ul#navigation a
{
    display: block;
    float: left;
}
```

“Плавающие” элементы и ширина

В данном уроке ширина “плавающих” элементов <a> не задается. Это позволяет каждой позиции списка иметь собственную ширину, определяемую числом символов и окружающим заполнением.

Тем не менее обычно такой подход не считается удачным. Гораздо лучше установить ширину для всех “плавающих” объектов (особенно если они связаны непосредственно с изображением, имеющим неявную ширину).

Если ширина не установлена, результаты могут быть непредсказуемыми. Теоретически “плавающий” элемент с неопределенной шириной должен сократиться до наиболее широкого своего элемента. Это может быть слово, предложение или даже один символ, следовательно, результаты могут отличаться для различных браузеров.

> В таком случае результаты приемлемы, поскольку практически во всех современных браузерах (в том числе Internet Explorer 5+, Netscape 6+, Opera 6+, Firefox и Safari) списки с переопределенными стилями отображаются нормально.

Далее с помощью объявления `padding` необходимо добавить немного заполнения. Вы можете использовать значения `.2em` для верхнего и нижнего заполнения и `1em` для левого и правого заполнения, как показано в листинге 16.9.

Листинг 16.9. Код CSS, устанавливающий заполнение

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
    background: #036;
    float: left;
    width: 100%;
}
ul#navigation li
{
    display: inline;
}
ul#navigation a
{
    display: block;
    float: left;
    padding: .2em 1em;
}
```

Чтобы убрать подчеркивание ссылок, используйте `text-decoration: none;`. Чтобы установить цвет текста и фона, используйте `color: #fff;` (белый) и `background: #036;`, как показано в листинге 16.10. При желании данные значения можно изменить.

Листинг 16.10. Код CSS, устанавливающий обрамление и цвет текста, а также цвет фона

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
    background: #036;
    float: left;
    width: 100%;
}
ul#navigation li
{
    display: inline;
}
ul#navigation a
{
    display: block;
    float: left;
    padding: .2em 1em;
    text-decoration: none;
    color: #fff;
    background: #036;
}
```

Чтобы разграничить элементы списка, в конце каждого элемента добавляется белая линия-разделитель. Для этого с помощью `border-right: 1px solid #fff;` справа от каждого элемента добавляется белая граница. Соответствующий код приводится в листинге 16.11, результат выполнения этого кода иллюстрируется на рис. 16.3.

Листинг 16.11. Код CSS, устанавливающий границу

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
    background: #036;
    float: left;
    width: 100%;
}
ul#navigation li
{
    display: inline;
}
ul#navigation a
{
    display: block;
    float: left;
    padding: .2em 1em;
    text-decoration: none;
    color: #fff;
    background: #036;
    border-right: 1px solid #fff;
}
```

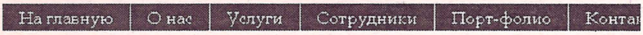


Рис. 16.3. Элемент <a> с переопределенным стилем

Определение стиля псевдокласса `:hover`

Для изменения стиля ссылок при визуальном выделении используется псевдокласс `:hover`. В данном случае цвет фона выбран равным `#69C`, а цвет текста — равным `#000`, как показано в листинге 16.12 и проиллюстрировано на рис. 16.4. При желании данные цвета можно изменить.

Листинг 16.12. Код CSS, задающий параметры визуального выделения

```
ul#navigation
{
    margin-left: 0;
    padding-left: 0;
    list-style-type: none;
    background: #036;
    float: left;
    width: 100%;
}
ul#navigation li
{
    display: inline;
}
ul#navigation a
{
    display: block;
    float: left;
    padding: .2em 1em;
    text-decoration: none;
    color: #fff;
    background: #036;
    border-right: 1px solid #fff;
}
ul#navigation a:hover
{
    color: #000;
    background: #69C;
}
```

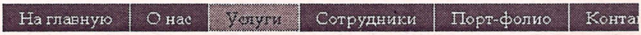
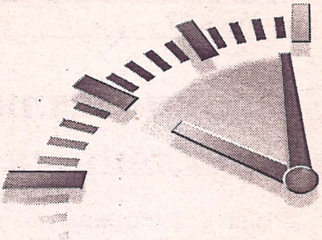

На главную
О нас
Услуги
Сотрудники
Порт-фолио
Конта

Рис. 16.4. Завершенный список

Резюме

Из этого урока вы узнали, что стиль списка можно определить так, чтобы результат выглядел, как горизонтальная панель навигации. Вы научились использовать “плавающие” элементы `<ul>` и `<a>` и псевдокласс `:hover`. В следующем уроке вы научитесь определять стиль рамок с закругленными углами.

Урок 17



Стилевое оформление рамки с закругленными углами

В этом уроке вы научитесь с помощью четырех изображений углов создавать растяжимую рамку с закругленными углами.

Подготовка HTML-кода

HTML-код этого урока состоит из общего контейнера `<div>`, заголовка и двух абзацев. Стиль элемента `<div>` определяется с помощью `pullquote` `id`, а стиль второго абзаца — с помощью класса `furtherinfo`, как показано в листинге 17.1.

Листинг 17.1. Код HTML, содержащий разметку рамки с закругленными углами

```
<div id="pullquote">
    <h2>Heading here</h2>
    <p>
        Lorem ipsum dolor sit amet....
    </p>
    <p class="furtherinfo">
        <a href="#">More information</a>
    </p>
</div>
```

Создание иллюзии закругленных углов

Создать растяжимую рамку с закругленными углами можно множеством методов. В методе, описываемом в этом уроке, используется четыре отдельных изображения углов.

Эти изображения не должны помещаться в код HTML, поскольку они являются исключительно средствами представления. В идеальном случае их необходимо применить как фоновое изображение, используя CSS.

Левое верхнее изображение будет применяться как фон к контейнеру `<div>`, а правое верхнее — к элементу `<h2>`.

Левое нижнее изображение будет применяться к последнему элементу `<p>` внутри рамки. Данный элемент `<p>` мы отнесем к классу `furtherinfo`, чтобы отличать его от других абзацев в рамке.

Правое нижнее изображение будет применяться к специальному экземпляру элемента `<a>`.

Объединение элементов для получения закругленных углов

Существует множество способов создания растяжимой рамки с закругленными углами.

Один из широко распространенных методов предусматривает вложение четырех уровней элементов `<div>`, которые затем применяются для расположения фоновых изображений.

Отметим, что по возможности лучше использовать существующие элементы или экземпляры элементов HTML, не создавая новые. Дополнительные элементы привносят ненужную разметку, которая может увеличить размер страницы и затруднить эксплуатацию кода.

Создание селекторов для определения стиля рамки с закругленными углами

Чтобы определить стиль рамки с закругленными углами, необходимо использовать пять селекторов, показанных в листинге 17.2.

Листинг 17.2. Код CSS, иллюстрирующий селекторы, которые используются для стилевого оформления структуры с двумя колонками

```
div#pullquote
div#pullquote h2
div#pullquote p
div#pullquote p.furtherinfo
div#pullquote p.furtherinfo a
```

Подготовка изображений

Четыре изображения, используемые в этом уроке, показаны на рис. 17.1–17.4. Стиль и цвет данных изображений вы можете изменить по своему усмотрению.

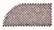


Рис. 17.1. Изображение 1, которое будет применяться к элементу `<div>`. Длина изображения должна составлять свыше 2000 пикселей, чтобы оно могло растягиваться до ширины экрана самого большого монитора

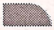


Рис. 17.2. Изображение 2, которое будет применяться к элементу `<h2>`

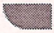


Рис. 17.3. Изображение 3, которое будет применяться к элементу `<p>`, стиль которого задан с использованием класса `.furtherinfo`

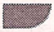


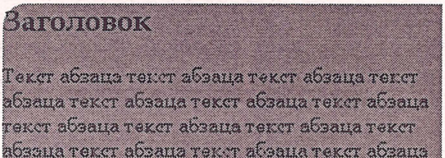
Рис. 17.4. Изображение 4, которое будет применяться к элементу <a>

Стилевое оформление элемента `<div>`

При создании рамки с закругленными углами первым этапом является стилевое оформление элемента `<div>`, вокруг которого со всех сторон необходимо добавить свободное пространство. Для этого можно использовать свойство `margin: 2em`. Далее элементу `<div>` требуется фоновое изображение, применяемое к левому верхнему углу. Как показано в листинге 17.3, для этого следует использовать свойство `background: #09f url(lesson17a.gif) no-repeat;`. Результаты выполнения данного кода представлены на рис. 17.5.

Листинг 17.3. Код CSS, определяющий стилевое оформление элемента `<div>`

```
div#pullquote
{
    margin: 2em;
    background: #09f url(lesson17a.gif) no-repeat;
}
```



Заголовок

Текст абзаца текст абзаца.

Подробнее

Рис. 17.5. Элемент <div> с переопределенным стилем

Стилевое оформление элемента <h2>

Итак, со стилем элемента `<div>` мы закончили. Теперь необходимо представить элемент `<h2>`, в правом верхнем углу которого будет располагаться второе фоновое изображение.

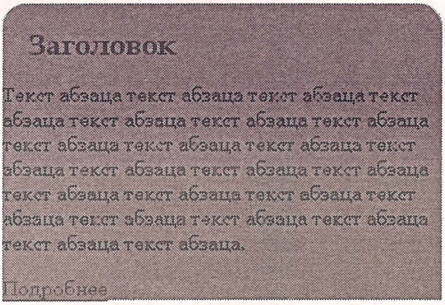
Поля элемента `<h2>` отключаются с использованием свойства `margin: 0`.

Далее элементу `<h2>` требуется заполнение на всех сторонах, исключая нижнюю. Для этого применяется свойство `padding: 20px 20px 0 20px`.

Наконец с помощью свойства `background: url (lesson17b.gif) no-repeat 100% 0;` к элементу `<h2>` добавляется фоновое изображение. Горизонтальное положение фонового изображения устанавливается равным `100%`, так что правая граница изображения будет располагаться рядом с правой границей элемента `<h2>`. Как показано в листинге 17.4, для изображения также устанавливается параметр `no-repeat`, поэтому оно не будет повторяться на фоне элемента `<h2>`. Результаты выполнения данного кода представлены на рис. 17.6.

Листинг 17.4. Код CSS, определяющий стилевое оформление элемента `<div>`

```
div#pullquote
{
    margin: 2em;
    background: #09f url(lesson17a.gif) no-repeat;
}
div#pullquote h2
{
    margin: 0;
    padding: 20px 20px 0 20px;
    background: url(lesson17b.gif) no-repeat 100% 0;
}
```



Заголовок

Текст абзаца текст абзаца.

Подробнее

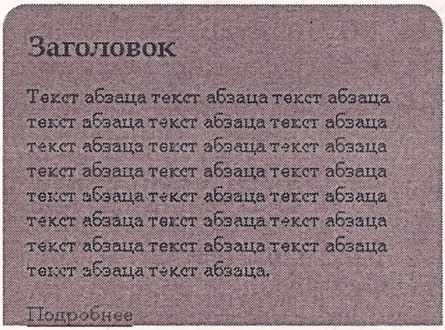
Рис. 17.6. Элемент <h2> с переопределенным стилем

Стилевое оформление элемента <p>

К элементу `<p>` необходимо с обеих сторон добавить заполнение, чтобы отделить его от границ контейнера. Для этого можно использовать свойство `padding: 0 20px`, как показано в листинге 17.5. Результаты выполнения данного кода представлены на рис. 17.7.

Листинг 17.5. Код CSS, определяющий стилевое оформление элемента `<p>`

```
div#pullquote
{
    margin: 2em;
    background: #09f url(lesson17a.gif) no-repeat;
}
div#pullquote h2
{
    margin: 0;
    padding: 20px 20px 0 20px;
    background: url(lesson17b.gif) no-repeat 100% 0;
}
div#pullquote p
{
    padding: 0 20px;
}
```



Заголовок

Текст абзаца текст абзаца.

Подробнее

Рис. 17.7. Элемент <p> с переопределенным стилем

Стилевое оформление класса `.furtherinfo`

Стиль последнего абзаца внутри рамки с закругленными углами задается с помощью класса `.furtherinfo`. С помощью данного абзаца в левом нижнем углу размещается третье изображение.

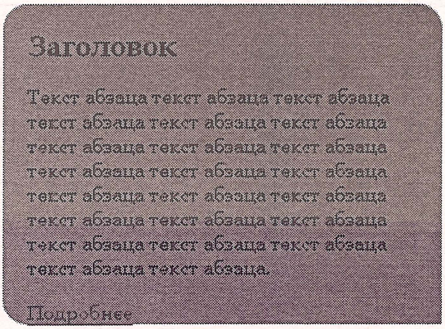
Для абзаца необходимо указать заполнение, которое, однако, можно задавать только для левой границы. Поэтому используем свойство `padding: 0 0 0 20px`.

Чтобы установить фоновое изображение, применяется свойство `background: url(lesson17c.gif) no-repeat 0 100%;`. Вертикальное положение устанавливается равным `100%`, поэтому нижняя граница изображения будет выровнена по нижней границе элемента <p>. Как показано в листинге 17.6, изображение также задается с параметром `no-repeat`, поэтому оно не будет повторяться на фоне элемента <p>. Результаты выполнения данного кода представлены на рис. 17.8.

Листинг 17.6. Код CSS, определяющий стилевое оформление класса `.furtherinfo`

```
div#pullquote
{
   margin: 2em;
```

```
    background: #09f url(lesson17a.gif) no-repeat;
}
div#pullquote h2
{
    margin: 0;
    padding: 20px 20px 0 20px;
    background: url(lesson17b.gif) no-repeat 100% 0;
}
div#pullquote p
{
    padding: 0 20px;
}
div#pullquote p.furtherinfo
{
    padding: 0 0 0 20px;
    background: url(lesson17c.gif) no-repeat 0 100%;
}
```



Заголовок

Текст абзаца текст абзаца.

Подробнее

Рис. 17.8. Класс `.furtherinfo` *с переопределенным стилем*

Стилевое оформление элемента <a>

Правое нижнее изображение применяется к элементу `<a>`.

Чтобы изображение было показано правильно, элемент `<a>` вначале необходимо перевести на блочный уровень, используя `display: block`.

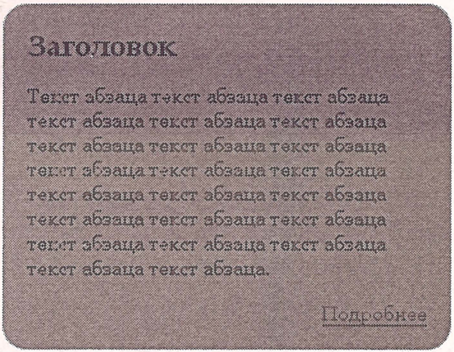
Следующий шаг заключается в добавлении заполнения с помощью свойства `padding: 0 20px 20px 0;`. Таким образом, будет увеличено заполнение справа и снизу от элемента.

Содержимое элемента `<a>` выравнивается по правому краю с помощью свойства `text-align: right`.

Как показано в листинге 17.7, с помощью свойства `background: url (lesson17d.gif) no-repeat 100% 100%;` к правой нижней границе элемента `<a>` применяется фоновое изображение. Результаты выполнения данного кода представлены на рис. 17.9.

Листинг 17.7. Код CSS, определяющий стилевое оформление элемента `<a>`

```
div#pullquote
{
    margin: 2em;
    background: #09f url(lesson17a.gif) no-repeat;
}
div#pullquote h2
{
    margin: 0;
    padding: 20px 20px 0 20px;
    background: url(lesson17b.gif) no-repeat 100% 0;
}
div#pullquote p
{
    padding: 0 20px;
}
div#pullquote p.furtherinfo
{
    padding: 0 0 0 20px;
    background: url(lesson17c.gif) no-repeat 0 100%;
}
div#pullquote p.furtherinfo a
{
    display: block;
    padding: 0 20px 20px 0;
    text-align: right;
    background: url(lesson17d.gif) no-repeat 100% 100%;
}
```



Заголовок

Текст абзаца текст абзаца.

Подробнее

Рис. 17.9. Элемент <a> с переопределенным стилем

Создание варианта с фиксированной длиной

Создание рамки с закругленными углами, имеющей фиксированную длину, требует всего двух фоновых изображений.

Первое изображение можно связать с контейнером `<div>`, используя свойство `background: #09f url(lesson17f.gif) no-repeat 0 100%;`. Таким образом, данное изображение, показанное на рис. 17.10, будет помещено в нижний угол рамки.

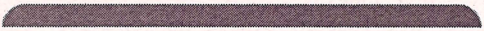


Рис. 17.10. Фоновое изображение, с помощью стиля привязываемое к элементу `<div>`

Ширина элемента `<div>` должна соответствовать ширине двух изображений; в данном случае выбрано значение `400px` (при желании вы можете выбрать любое другое).

Кроме того, для элемента `<div>` необходимо задать заполнение, чтобы текст не сливался с верхней границей изображения. Для этого применяется свойство `padding-bottom: 20px`.

Для элемента `<p>` необходимо задать заполнение по бокам, чтобы отодвинуть его от границ контейнера. Для этого используется свойство `padding: 0 20px`.

Элемент <h2> должен иметь нулевые поля, чтобы между верхней границей контейнера и изображением не было никаких промежутков. Кроме того, требуется ввести заполнение сверху, справа и слева, используя свойство `padding: 20px 20px 0 20px;`.

Чтобы применить фоновое изображение к элементу <h2>, используется свойство `background: url(lesson17e.gif) no-repeat 100% 0;`. Таким образом, изображение, показанное на рис. 17.11, располагается в левом верхнем углу элемента.

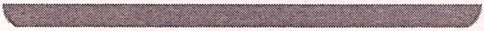
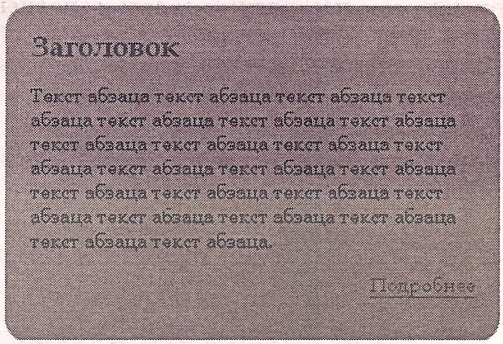


Рис. 17.11. Фоновое изображение, связываемое с элементом <h2>

Содержимое абзаца `.furtherinfo` необходимо выровнять по правому краю. Для этого переопределяется свойство `text-align: right`, как показано в листинге 17.8. Результаты выполнения данного кода представлены на рис. 17.12.

Листинг 17.8. Код CSS, определяющий стилевое оформление закругленной рамки фиксированной ширины

```
div#pullquote
{
    background: #09f url(lesson17f.gif) no-repeat 0 100%;
    width: 400px;
    padding-bottom: 20px;
}
div#pullquote p
{
    padding: 0 20px;
}
div#pullquote h2
{
    margin: 0;
    padding: 20px 20px 0 20px;
    background: url(lesson17e.gif) no-repeat 100% 0;
}
div#pullquote p.furtherinfo
{
    text-align: right;
}
```



Заголовок

Текст абзаца текст абзаца.

Подробнее

Рис. 17.12. Рамка с закругленными углами и фиксированной шириной

Создание гибкого варианта только с верхним изображением

Чтобы создать гибкий вариант рамки с закругленным углами, содержащий только верхнюю часть изображения, требуется два фоновых рисунка.

Первое изображение можно применить к контейнеру `<div>`, используя `background: #fff url(lesson17g.jpg) no-repeat;`. Таким образом, изображение, показанное на рис. 17.13, будет помещено в левом верхнем углу рамки.

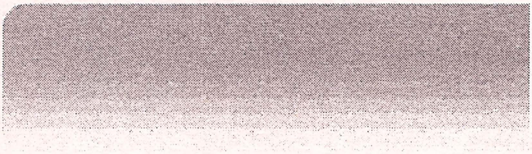


Рис. 17.13. Фоновое изображение, которое будет применяться к элементу `<div>`. Длина изображения должна составлять свыше 2000 пикселей, чтобы оно могло растягиваться до ширины экрана самого большого монитора

Для элемента `<p>` необходимо задать заполнение по бокам, чтобы отодвинуть его от границ контейнера. Для этого можно использовать свойство `padding: 0 20px`.

Поля элемента `<h2>` должны быть нулевыми, чтобы между верхней границей контейнера и изображением не было никаких промежутков. Кроме того, требуется ввести заполнение сверху, справа и слева, используя свойство `padding: 20px 20px 0 20px;`.

Чтобы применить фоновое изображение к элементу `<h2>`, используется свойство `background: url(lesson17h.gif) no-repeat 100% 0;`. Таким образом, изображение, показанное на рис. 17.14, располагается в правом верхнем углу элемента.

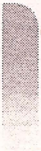


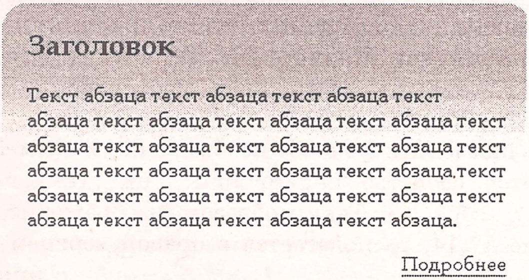
Рис. 17.14. Фоновое изображение элемента <h2>

Наконец содержимое абзаца `.furtherinfo` необходимо выровнять по правому краю. Для этого переопределяется свойство `text-align: right`, как показано в листинге 17.9. Результаты выполнения данного кода представлены на рис. 17.15.

Листинг 17.9. Код CSS, определяющий стилевое оформление варианта только с верхним изображением

```
div#pullquote
{
    background: #fff url(lesson17g.jpg) no-repeat;
}
div#pullquote p
{
    padding: 0 20px;
}
div#pullquote h2
{
    margin: 0;
    padding: 20px 20px 0 20px;
    background: url(lesson17h.jpg) no-repeat 100% 0;
}
```

```
div#pullquote p.furtherinfo
{
    text-align: right;
}
```



Заголовок

Текст абзаца текст абзаца.

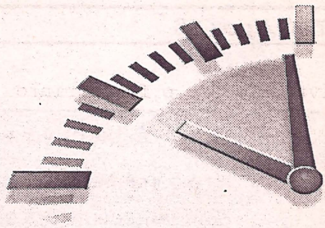
Подробнее

Рис. 17.15. *Стиль варианта только с верхним изображением*

Резюме

Из этого урока вы узнали, как, применяя изображения к четырем различным элементам, создать иллюзию гибкой рамки с закругленными углами. Кроме того, вы научились создавать рамку с закругленными углами фиксированной ширины и рамку, состоящую только из верхнего изображения. В следующем уроке вы научитесь настраивать стиль заголовка сайта, используя элемент <h1>, изображение и список.

Урок 18



Создание заголовка сайта

В этом уроке вы научитесь устанавливать стиль заголовка сайта, используя элемент <h1>, изображение и список.

Подготовка HTML-кода

Как показано в листинге 18.1, HTML-код этого урока состоит из трех основных компонентов: элемента `<div>`, помогающего семантически определить заголовок, элемента `<h1>` и элемента `<ul>`, используемого для навигации.

Листинг 18.1. Код HTML, содержащий разметку заголовка сайта

```
<div id="container">
    <h1>
        <a href="/"><img src="header.jpg" alt="Sitename"a></a>
    </h1>
    <ul id="topnav">
        <li><a href="#">Skip to content</a></li>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Staff</a></li>
        <li><a href="#">Portfolio</a></li>
        <li><a href="#">Contact</a></li>
    </ul>
</div>
```

Ссылка “пропустить”

Что означает ссылка “пропустить”?

Данная ссылка позволяет пользователям пропустить ссылки навигации и сразу обратиться к основному содержимому страницы.

Подобные ссылки очень удобны для слепых, людей с плохим зрением и тех, кто использует текстовые браузеры, мобильные телефоны и КПК.

Некоторые разработчики используют скрытые ссылки, специально созданные для слепых пользователей. Тем не менее эти ссылки должны быть видимы и для людей, которые не могут использовать мышь и должны для навигации по сайту применять табуляцию.

Создание селекторов для определения стиля заголовка

Чтобы переопределить стиль заголовка сайта, потребуется 10 селекторов, показанных в листинге 18.2.

Листинг 18.2. Код CSS, иллюстрирующий селекторы, которые используются для определения стиля заголовка

```
body {...}
#container {...}
h1 {...}
h1 img {...}
ul#topnav {...}
ul#topnav li {...}
ul#topnav li a:link {...}
ul#topnav li a:visited {...}
ul#topnav li a:hover {...}
ul#topnav li a:active {...}
```

Стилевое оформление элемента <body>

Одни браузеры для отделения содержимого от границ окна браузера используют в элементе <body> поля, другие — заполнение. Поскольку заголовок данного сайта располагается вверху окна браузера, и поля, и заполнение следует сделать нулевыми.

Теоретически, чтобы отцентрировать контейнер на странице, можно применить автоматически заданные поля слева и справа. Однако некоторые браузеры не позволяют отцентрировать контейнер таким способом, поскольку игнорируют автоматические поля. Данную проблему можно решить, добавив два простых объявления.

Первое объявление, `text-align: center`, применяется к элементу <body>. Второе, `text-align: left`, добавляется к набору правил контейнера (см. следующий раздел, "Стилевое оформление контейнера", этого урока).

Кроме того, для элемента <body> необходимо установить свойства `background-color` и `color`. В данном случае цвет фона выбирается равным `#B0BFC2`, а цвет текста — равным `#444`, как показано в листинге 18.3. Результаты выполнения данного кода представлены на рис. 18.1.

Листинг 18.3. Код CSS, определяющий стилевое оформление элемента <body>

```
body
{
    margin: 0;
    padding: 0;
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
```

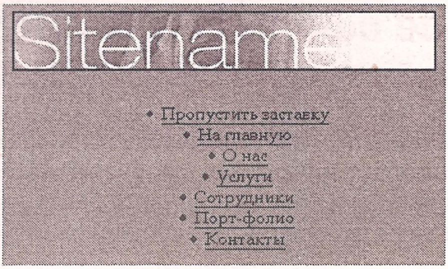


Рис. 18.1. Элемент <body> с переопределенным стилем

Стилевое оформление контейнера

Теперь, завершив стилевое оформление элемента `<body>`, мы расположили все содержимое по центру страницы. Данные настройки можно изменить, задав для контейнера свойство `text-align: left`.

После этого для браузеров, поддерживающих автоматические поля, применяется правильный метод центрирования: `margin: 0 auto`.

Ширину контейнера можно задать равной 700px (вы можете изменить это значение по своему усмотрению).

Наконец, как показано в листинге 18.4, для контейнера необходимо задать белый фон, используя свойство `background: #fff`. Результаты выполнения данного кода представлены на рис. 18.2.

Листинг 18.4. Код CSS, определяющий стилевое оформление контейнера

```
body
{
    margin: 0;
    padding: 0;
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF;
}
```

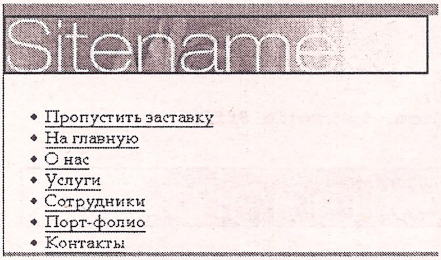


Рис. 18.2. Контейнер с переопределенным стилем

Стилевое оформление элемента <h1>

Как было показано в листинге 18.1, элемент `<h1>` содержит изображение. Данный рисунок помещается внутрь элемента `<h1>` для придания ему большего семантического значения. Программы чтения с экрана и текстовые браузеры прочтут атрибут `"Sitename"`, просто как заголовок текста.

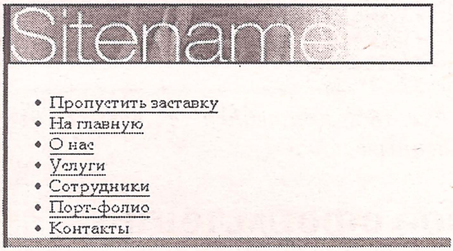
Поля и заполнение необходимо установить равными `0`, чтобы изображение располагалось в самом верху окна браузера.

Кроме того, можно добавить белую линию к нижней границе элемента `<h1>`, используя сокращенную форму записи `border: 1px solid #fff;`, как показано в листинге 18.5. Результаты выполнения данного кода представлены на рис. 18.3.

Листинг 18.5. Код CSS, определяющий стилевое оформление элемента `<h1>`

```
body
{
    margin: 0;
    padding: 0;
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
#container
```

```
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF;
}
h1
{
    margin: 0;
    padding: 0;
    border-bottom: 1px solid #fff;
}
```



Sitename

- Пропустить заставку
- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

Рис. 18.3. Элемент <h1> с переопределенным стилем

Стилевое оформление элемента <image>

Изображение должно задаваться с атрибутом `display: block`. Таким образом, мы удаляем промежутки между ним и верхней границей элемента <h1>.

Изображение вкладывается внутрь элемента <a>. В некоторых браузерах это приведет к тому, что изображение будет показано с границей шириной 2 пикселя. Чтобы избежать этого, в стиль изображения следует включить определение `border: 0`, как показано в листинге 18.6. Результаты выполнения данного кода представлены на рис. 18.4.

Листинг 18.6. Код CSS, определяющий стилевое оформление элемента <image>

```
body
{
    margin: 0;
    padding: 0;
    text-align: center;
    font: 85% arial, helvetica, sans-serif;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF;
}
h1
{
    margin: 0;
    padding: 0;
    border-bottom: 1px solid #fff;
}
h1 img
{
    display: block;
    border: 0;
}
```

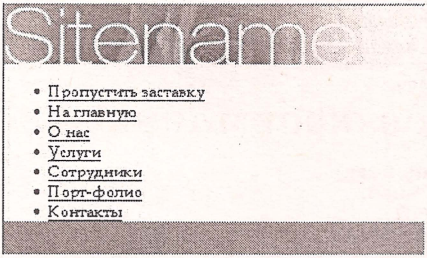

Sitename
Пропустить заставку
На главную
О нас
Услуги
Сотрудники
Порт-фолио
Контакты

Рис. 18.4. Элемент <image> с переопределенным стилем

Стилевое оформление элемента `<ul>`

Как обсуждалось в уроке 15, “Создание вертикальной панели навигации”, большинство браузеров отображает списки HTML с отступом слева. Чтобы убрать этот отступ для всех браузеров, необходимо переопределить и поля, и заполнение.

В данном случае мы устанавливаем поля равными 0, чтобы списки выравнивались по элементу `<h1>`.

К позициям списка необходимо добавить заполнение `5px` сверху и `10px` слева и справа. Для этого можно использовать сокращенную форму записи `padding: 5px 10px`.

Маркеры HTML необходимо удалить с помощью `list-style-type: none`.

В стиль элемента `<ul>` включается объявление `background-color: #387A9B`.

Поскольку нам не нужно, чтобы данные правила применялись ко всем элементам `<ul>` страницы, следует использовать селектор-потомок, отбирающий только список `topnav`. Для этого применяется `ul#topnav`, как показано в листинге 18.7. Результаты выполнения данного кода представлены на рис. 18.5.

Листинг 18.7. Код CSS, определяющий стилевое оформление элемента `<ul>`

```
body
{
    margin: 0;
    padding: 0;
    text-align: center;
    font: 85% arial, helvetica, sans-serif;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF;
}
h1
{
    margin: 0;
    padding: 0;
    border-bottom: 1px solid #fff;
}
h1 img
{
    display: block;
    border: 0;
}
ul#topnav
{
    margin: 0;
    padding: 5px 10px;
    list-style-type: none;
    background: #387A9B;
}
```

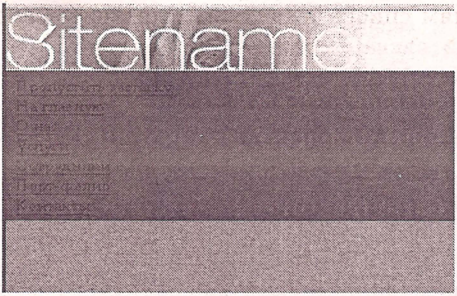


Рис. 18.5. Элемент <ul> с переопределенным стилем

Стилевое оформление элемента `<li>`

Наш список должен отображаться на экране горизонтально, а не вертикально. Для этого в стиле элемента `<li>` задается `display: inline`.

После этого в `<li>` необходимо добавить графические маркеры. Лучше всего это сделать, используя фоновое изображение, установленное с помощью `background: url(header-bullet.gif) no-repeat 0 50%;`. Таким образом, рядом с каждым элементом списка будет расположено одно изображение, отцентрированное по вертикали.

Чтобы отодвинуть текст от фонового изображения, потребуется заполнение. В данном случае будет использовано свойство `padding: 0 10px 0 8px;`, как показано в листинге 18.8. Таким образом, 10 пикселей заполнения будет добавлено справа, а 8 пикселей — слева от каждого элемента списка. Результаты выполнения данного кода представлены на рис. 18.6.

Листинг 18.8. Код CSS, определяющий стилевое оформление элемента `<li>`

```
body
{
    margin: 0;
    padding: 0;
    text-align: center;
    font: 85% arial, helvetica, sans-serif;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF;
}
h1
{
    margin: 0;
    padding: 0;
    border-bottom: 1px solid #fff;
}
h1 img
{
    display: block;
    border: 0;
}
ul#topnav
{
    margin: 0;
    padding: 5px 10px;
    list-style-type: none;
    background: #387A9B;
}
ul#topnav li
{
    display: inline;
    background: url(header-bullet.gif) no-repeat 0 50%;
    padding: 0 10px 0 8px;}
}
```

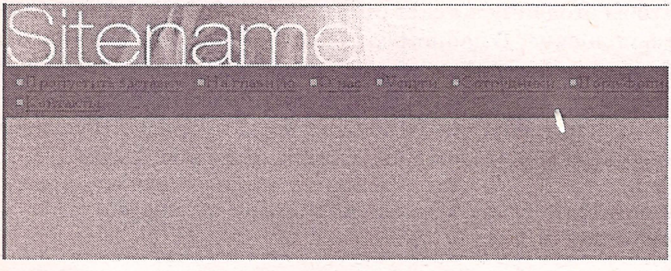


Рис. 18.6. Элемент `<li>` с переопределенным стилем

Стилевое оформление элемента <a>

Чтобы не затронуть все ссылки на странице, следует использовать определенные селекторы, в данном случае — `ul#topnav li a:link`, `ul#topnav li a:visited`, `ul#topnav li a:hover` и `ul#topnav li a:active`.

Псевдоклассы ссылки и посещенной ссылки можно задать с помощью свойств `text-decoration: none` (отключает подчеркивание ссылки) и `color: #FFF`, как показано в листинге 18.9.

Псевдоклассы визуального выделения и активной ссылки задаются со свойствами `text-decoration: none`, `color: #387A9B;` и `background: #FFF;`. Результаты выполнения данного кода представлены на рис. 18.7.

Листинг 18.9. Код CSS, определяющий стилевое оформление элемента <a>

```
body
{
    margin: 0;
    padding: 0;
    text-align: center;
    font: 85% arial, helvetica, sans-serif;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF;
}
h1
{
    margin: 0;
    padding: 0;
    border-bottom: 1px solid #fff;
}
h1 img
{
    display: block;
    border: 0;
}
ul#topnav
{
    margin: 0;
    padding: 5px 10px;
    list-style-type: none;
    background: #387A9B;
}
ul#topnav li
{
    display: inline;
    background: url(header-bullet.gif) no-repeat 0 50%;
    padding: 0 10px 0 8px;
}
ul#topnav li a:link, ul#topnav li a:visited
{
    text-decoration: none;
    color: #fff;
}
ul#topnav li a:hover, ul#topnav li a:active
{
    text-decoration: none;
    color: #387A9B;
    background: #fff;
}
```

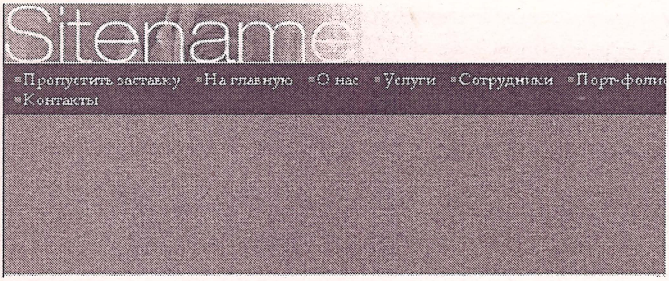
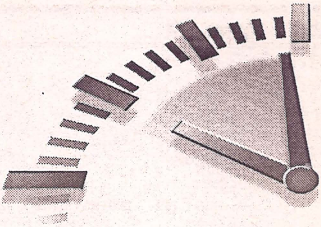

Sitename
Пропустить заставку
На главную
О нас
Услуги
Сотрудники
Порт-фоли
Контакты

Рис. 18.7. Элемент <a> с переопределенным стилем

Резюме

Из этого урока вы узнали, как в окне браузера выравнивать содержимое по центру и определять стили элементов <h1>, <image> и <ul> в виде ссылок. В следующем уроке вы научитесь размещать страницу с двумя колонками с заголовком и примечанием.

Урок 19



Расположение двух колонок с заголовком и примечанием

В этом уроке вы научитесь размещать страницу из двух колонок с заголовком и примечанием. Существует множество способов размещения данных элементов. Метод, рассмотренный в данном уроке, включает применение “плавающих” объектов, поскольку это наиболее надежный подход, приемлемый для большинства современных браузеров.

Подготовка HTML-кода

HTML-код этого урока состоит из пяти основных контейнеров: элемента `<h1>` и трех элементов `<div>`, расположенных внутри другого контейнера `<div>`, как показано в листинге 19.1.

Листинг 19.1. Код HTML, содержащий разметку для урока 19

```
<div id="container">
    <h1>
        Sitename
    </h1>
    <div id="nav">
        <ul>
          <li><a href="#">Home</a></li>
          <li><a href="#">About us</a></li>
          <li><a href="#">Services</a></li>
          <li><a href="#">Staff</a></li>
          <li><a href="#">Portfolio</a></li>
          <li><a href="#">Contact us</a></li>
        </ul>
    </div>
    <div id="content">
        <h2>
          About Sitename
        </h2>
        <p>
          Lorem ipsum dolor....
        </p>
        <p>
          Ut wisi enim ad...
        </p>
    </div>
    <div id="footer">
        Copyright &copy; Sitename 2005
    </div>
</div>
```

Создание селекторов для стилевого оформления двух колонок

Чтобы определить стиль двух колонок и их содержимого, потребуется 12 селекторов, показанных в листинге 19.2.

Листинг 19.2. Код CSS, иллюстрирующий селекторы, которые применяются для стилевого оформления двух колонок

```
body {...}
#container {...}
h1 {...}
#nav {...}
#nav ul {...}
#nav li {...}
#content {...}
#footer {...}
h2 {...}
a:link {...}
a:visited {...}
a:hover, a:active {...}
```

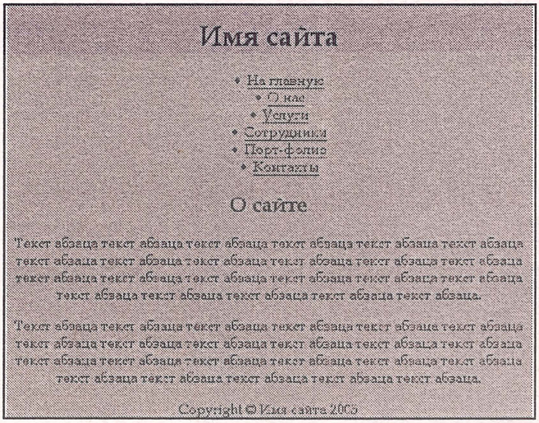
Стилевое оформление элемента <body>

Как обсуждалось в уроке 18, “Создание заголовка сайта”, для выравнивания контейнера по центру страницы необходимы решения, подходящие для всех браузеров, не поддерживающих автоматически заданные поля. Первое, что необходимо для этого сделать, — применить к элементу `<body>` свойство `text-align: center`, как показано в листинге 19.3. Результаты представлены на рис. 19.1.

Кроме того, для элемента `<body>` необходимо задать свойства `background-color` и `color`. Цвет фона, например, можно выбрать равным `#B0BFC2`, а цвет текста — равным `#444`.

Листинг 19.3. Код CSS, определяющий стилевое оформление элемента `<body>`

```
body
{
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
```



Имя сайта

- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

О сайте

Текст абзаца текст абзаца.

Текст абзаца текст абзаца.

Copyright © Имя сайта 2005

Рис. 19.1. Элемент <body> с переопределенным стилем

Стилевое оформление контейнера

Определив стиль элемента `<body>`, получаем выровненное по центру все содержимое страницы. Чтобы исправить положение, зададим для контейнера свойство `text-align: left`.

После этого для браузеров, поддерживающих автоматически заданные поля, устанавливается правильный метод центрирования: `margin: 0 auto`.

Ширину контейнера можно установить равной `700px` (вы можете использовать любое удобное для вас значение).

Наконец для контейнера необходимо задать фоновое изображение, используя `background: #FFF url(header-base.gif) repeat-y;`, как показано в листинге 19.4. Фоновое изображение представлено на рис. 19.2, а внешний вид полученной страницы показан на рис. 19.3.

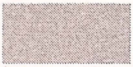


Рис. 19.2. Фоновое изображение контейнера

Создание иллюзии цветов колонок

Одна из проблем, связанных с использованием "плавающих" контейнеров, заключается в том, что в общем случае их глубины достаточно для помещения в них содержимого. Если одна колонка гораздо короче другой, создать колонки, используя только цвета фона, может быть затруднительно.

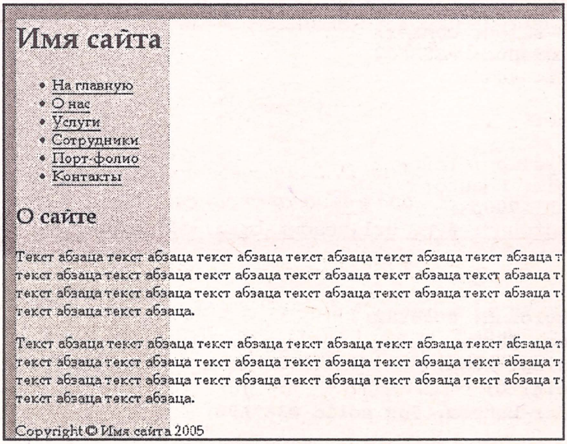
Как же тогда сделать так, чтобы цвет фона более короткой колонки "дотягивался" до низа страницы?

Одно простое решение заключается в использовании фонового изображения, создающего иллюзию цвета колонки. Данное изображение можно добавить к общему контейнеру в качестве фонового и повторить вниз по оси y.

В результате "плавающие" контейнеры располагаются поверх данного повторяющегося изображения и цвета распространяются до самого низа страницы независимо от того, какая колонка длиннее.

Листинг 19.4. Код CSS, определяющий стилевое оформление контейнера

```
body
{
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF url(header-base.gif) repeat-y;
}
```



Имя сайта

- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

О сайте

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца т
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца т
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца т
текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца т
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца т
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца т
текст абзаца текст абзаца.

Copyright © Имя сайта 2005

Рис. 19.3. Контейнер с переопределенным стилем

Стилевое оформление элемента <h1>

Первым шагом при определении стиля заголовка является задание цвета фона. В нашем случае применяется свойство `background: #D36832`. Цвет текста устанавливается равным `#fff`.

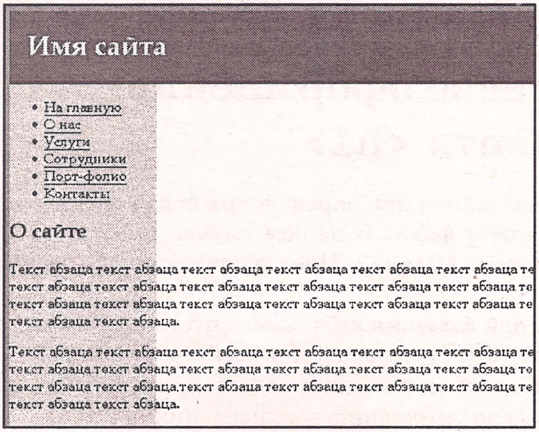
Далее для создания небольшого пустого пространства вокруг содержимого <h1> используется свойство `padding: 20px`.

После этого необходимо задать `margin: 0`, удаляя установленные по умолчанию верхнее и нижнее поля.

Кроме того, можно установить границу сверху от элемента `<h1>`, используя сокращенную запись `border: 5px solid #387A9B;`, как показано в листинге 19.5. Результаты выполнения данного кода представлены на рис. 19.4.

Листинг 19.5. Код CSS, определяющий стилевое оформление элемента `<h1>`

```
body
{
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF url(header-base.gif) repeat-y;
}
h1
{
    background: #D36832;
    color: #FFF;
    padding: 20px;
    margin: 0;
    border-bottom: 5px solid #387A9B;
}
```



Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца те
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца те
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца те
текст абзаца текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца те
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца те
текст абзаца текст абзаца.текст абзаца текст абзаца текст абзаца текст абзаца те
текст абзаца текст абзаца.

Рис. 19.4. *Элемент <h1> с переопределенным стилем*

Стилевое оформление контейнера #nav

Чтобы разместить контейнер `#nav` и элемент `#content` рядом, их необходимо сделать “плавающими”.

Чтобы сделать контейнер `#nav` “плавающим”, используйте свойство `float: left`. Кроме того, для этого элемента необходимо установить ширину, например, `130px`.

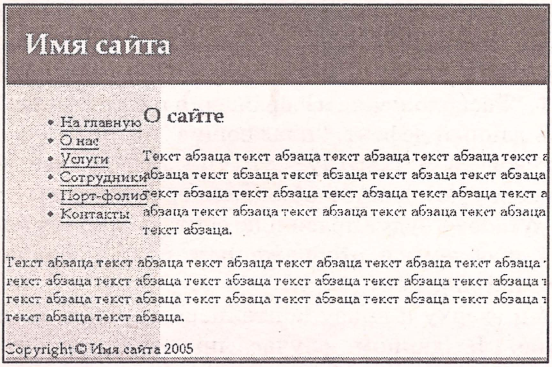
Отметим, что Internet Explorer 5 и 5.5 для Windows при определенных обстоятельствах отображает поля, удваивая заданный для них размер. Данный дефект браузера объясняется в уроке 11, “Расположение изображения и его подписи”. Чтобы исправить данный дефект, “плавающие” элементы следует задавать со свойством `display: inline`. Все остальные браузеры проигнорируют данное объявление, а Internet Explorer 5 и 5.5 для Windows будут правильно считать ширину поля.

Решив проблему с отображением полей, можно задать свойство `margin-left: 20px`.

Наконец вверху и внизу контейнера необходимо добавить заполнение. В данном случае применяется объявление `padding: 15px 0`, как показано в листинге 19.6. Результаты выполнения данного кода представлены на рис. 19.5.

Листинг 19.6. Код CSS, определяющий стилевое оформление контейнера `#nav`

```
body
{
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF url(header-base.gif) repeat-y;
}
h1
{
    background: #D36832;
    color: #FFF;
    padding: 20px;
    margin: 0;
    border-bottom: 5px solid #387A9B;
}
#nav
{
    float: left;
    width: 130px;
    display: inline;
    margin-left: 20px;
    padding: 15px 0;
}
```



Имя сайта

- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

О сайте

Текст абзаца текст абзаца текст абзаца текст абзаца текст а
абзаца текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца текст абзаца текст абзаца текст абзаца текст а
абзаца текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца текст абзаца.

Copyright © Имя сайта 2005

Рис. 19.5. Элемент #nav с переопределенным стилем

Стилевое оформление элемента `<ul>`

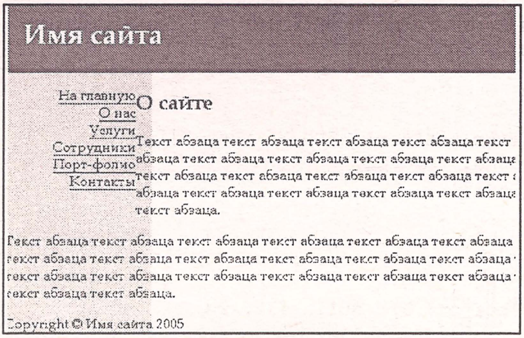
Чтобы убрать установленные в браузере значения по умолчанию, и поля, и заполнение необходимо обнулить.

Для удаления маркеров списка используйте свойство `list-style-type: none`.

Далее позиции списка необходимо выровнять по правому краю контейнера #nav. Для этого можно использовать объявление `text-align: right`, как показано в листинге 19.7. Результаты выполнения данного кода представлены на рис. 19.6.

Листинг 19.7. Код CSS, определяющий стилевое оформление элемента `<ul>`

```
body
{
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF url(header-base.gif) repeat-y;
}
h1
{
    background: #D36832;
    color: #FFF;
    padding: 20px;
    margin: 0;
    border-bottom: 5px solid #387A9B;
}
#nav
{
    float: left;
    width: 130px;
    display: inline;
    margin-left: 20px;
    padding: 15px 0;
}
#nav ul
{
    margin: 0;
    padding: 0;
    list-style-type: none;
    text-align: right;
}
```



Имя сайта

На главную
О нас
Услуги
Сотрудники
Порт-фолио
Контакты

О сайте

Текст абзаца текст абзаца текст абзаца текст абзаца текст
абзаца текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца текст абзаца текст абзаца текст абзаца текст
абзаца текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца.

екст абзаца текст абзаца текст абзаца текст абзаца текст абзаца
екст абзаца текст абзаца текст абзаца текст абзаца текст абзаца
екст абзаца текст абзаца текст абзаца текст абзаца текст абзаца
екст абзаца текст абзаца.

opyright © Имя сайта 2005

Рис. 19.6. Элемент `<ul>` *с переопределенным стилем*

Стилевое оформление элемента `<li>`

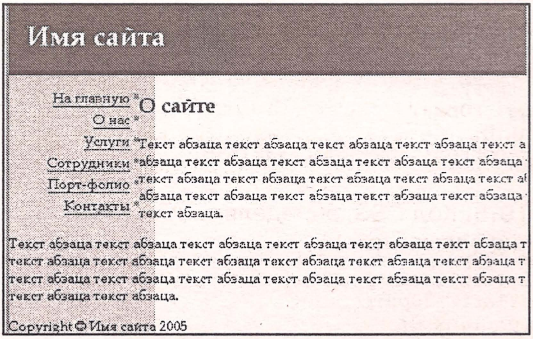
Теперь элемент списка получил фоновое изображение, которое будет выступать как маркер списка. Чтобы разместить фоновое изображение по правой границе элемента `<li>` на расстоянии `.4em` от его вершины, используйте объявление `background: url(header-bullet.gif) no-repeat 100% .4em;`. Кроме того, для изображения устанавливается свойство `no-repeat`, поэтому оно не будет повторяться по всему элементу `<li>`.

Далее к правой и нижней границам элемента `<li>` можно добавить заполнение, как показано в листинге 19.8. Заполнение справа отодвинет содержимое списка от края, чтобы оно не располагалось выше фонового изображения. Заполнение снизу обеспечивает некоторый промежуток между элементами списка. Результаты выполнения данного кода представлены на рис. 19.7.

Листинг 19.8. Код CSS, определяющий стилевое оформление элемента `<li>`

```
body
{
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
```

```
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF url(header-base.gif) repeat-y;
}
h1
{
    background: #D36832;
    color: #FFF;
    padding: 20px;
    margin: 0;
    border-bottom: 5px solid #387A9B;
}
#nav
{
    float: left;
    width: 130px;
    display: inline;
    margin-left: 20px;
    padding: 15px 0;
}
#nav ul
{
    margin: 0;
    padding: 0;
    list-style-type: none;
    text-align: right;
}
#nav li
{
    background: url(header-bullet.gif) no-repeat 100% .4em;
    padding: 0 10px 5px 0;
}
```



Имя сайта

На главную
О нас
Услуги
Сотрудники
Порт-фолио
Контакты

О сайте

Текст абзаца текст абзаца текст абзаца текст абзаца текст а
абзаца текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца текст абзаца текст абзаца текст абзаца текст а
абзаца текст абзаца текст абзаца текст абзаца текст абзаца
текст абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца т
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца т
текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца т
текст абзаца текст абзаца.

Copyright © Имя сайта 2005

Рис. 19.7. *Элемент `<li>` с переопределенным стилем*

Стилевое оформление контейнера `#content`

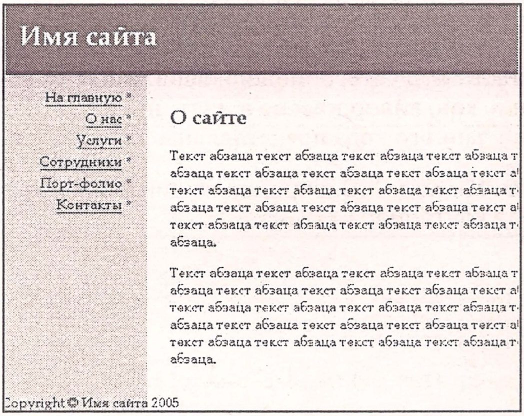
Контейнер `#content` требуется задать со свойством `float: left`, чтобы он располагался рядом с контейнером `#nav`. Кроме того, для него необходимо указать ширину, которая в данном случае равна `475px`.

Чтобы создать промежуток между двумя колонками, используйте свойство `margin-left: 45px`.

Наконец вверху и внизу контейнера необходимо добавить заполнение, используя `padding: 15px 0`, как показано в листинге 19.9. Результаты выполнения данного кода представлены на рис. 19.8.

Листинг 19.9. Код CSS, определяющий стилевое оформление контейнера `#content`

```
body
{
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF url(header-base.gif) repeat-y;
}
h1
{
    background: #D36832;
    color: #FFF;
    padding: 20px;
    margin: 0;
    border-bottom: 5px solid #387A9B;
}
#nav
{
    float: left;
    width: 130px;
    display: inline;
    margin-left: 20px;
    padding: 15px 0;
}
#nav ul
{
    margin: 0;
    padding: 0;
    list-style-type: none;
    text-align: right;
}
#nav li
{
    background: url(header-bullet.gif) no-repeat 100% .4em;
    padding: 0 10px 5px 0;
}
#content
{
    float: left;
    width: 475px;
    margin-left: 45px;
    padding: 15px 0;
}
```



Имя сайта

На главную *
О нас *
Услуги *
Сотрудники *
Порт-фолио *
Контакты *

О сайте

Текст абзаца текст абзаца текст абзаца текст абзаца т
абзаца текст абзаца текст абзаца текст абзаца текст а
текст абзаца текст абзаца текст абзаца текст абзаца т
абзаца текст абзаца текст абзаца текст абзаца текст а
текст абзаца текст абзаца текст абзаца текст абзаца т
абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца т
абзаца текст абзаца текст абзаца текст абзаца текст а
текст абзаца текст абзаца текст абзаца текст абзаца т
абзаца текст абзаца текст абзаца текст абзаца текст а
текст абзаца текст абзаца текст абзаца текст абзаца т
абзаца.

opyright © Имя сайта 2005

Рис. 19.8. *Элемент `#content` с переопределенным стилем*

Стилевое оформление контейнера `#footer`

Контейнер `#footer` отображается после контейнеров `#nav` и `#content`. Поскольку последние два контейнера являются “плавающими”, существует вероятность того, что контейнер `#footer` попытается разместиться рядом с ними. Чтобы предупредить эту возможность, при определении контейнера `#footer` используйте свойство `clear`. Допускаются следующие варианты: `clear: left`, `clear: right`, `clear: both` и `clear: none`.

В данном случае используется значение `clear: both`, применяя которое, мы указываем контейнеру `#footer` расположиться ниже двух “плавающих” контейнеров.

Чтобы задать цвет фона, используйте свойство `background: #387A9B`. Цвет текста можно установить равным `#fff`.

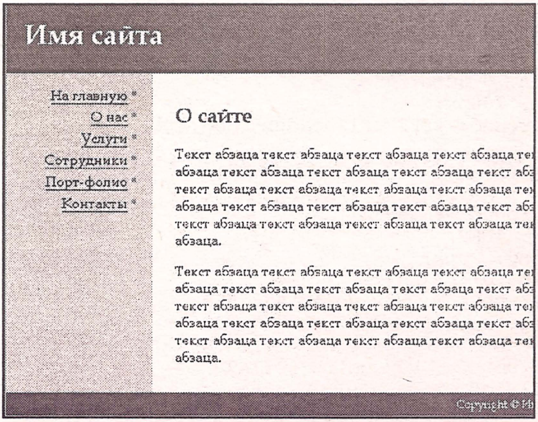
Далее для создания небольшого промежутка вокруг содержимого можно ввести заполнение 5px снизу и сверху, 10px слева и справа, используя, как показано в листинге 19.10, свойство `padding: 5px 10px`.

Чтобы выровнять содержимое примечания по правому краю, используйте `text-align: right`.

Наконец можно уменьшить размер примечания, поскольку обычно представляемая им информация менее важна, чем основной тест. Для этого можно использовать объявление `font-size: 80%`, как показано в листинге 19.10. Результаты выполнения данного кода представлены на рис. 19.9.

Листинг 19.10. Код CSS, определяющий стилевое оформление контейнера `#footer`

```
body
{
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF url(header-base.gif) repeat-y;
}
h1
{
    background: #D36832;
    color: #FFF;
    padding: 20px;
    margin: 0;
    border-bottom: 5px solid #387A9B;
}
#nav
{
    float: left;
    width: 130px;
    display: inline;
    margin-left: 20px;
    padding: 15px 0;
}
#nav ul
{
    margin: 0;
    padding: 0;
    list-style-type: none;
    text-align: right;
}
#nav li
{
    background: url(header-bullet.gif) no-repeat 100% .4em;
    padding: 0 10px 5px 0;
}
#content
{
    float: left;
    width: 475px;
    margin-left: 45px;
    padding: 15px 0;
}
#footer
{
    clear: both;
    background: #387A9B;
    color: #fff;
    padding: 5px 10px;
    text-align: right;
    font-size: 80%;
}
```



О сайте

Текст абзаца текст абзаца текст абзаца текст абзаца те
абзаца текст абзаца текст абзаца текст абзаца текст аб
текст абзаца текст абзаца текст абзаца текст абзаца те
абзаца текст абзаца текст абзаца текст абзаца текст аб
текст абзаца текст абзаца текст абзаца текст абзаца те
абзаца.

Текст абзаца текст абзаца текст абзаца текст абзаца те
абзаца текст абзаца текст абзаца текст абзаца текст аб
текст абзаца текст абзаца текст абзаца текст абзаца те
абзаца текст абзаца текст абзаца текст абзаца текст аб
текст абзаца текст абзаца текст абзаца текст абзаца те
абзаца.

Рис. 19.9. Элемент #footer с переопределенным стилем

Стилевое оформление элемента <h2>

Элемент <h2> используется для оформления основного заголовка страницы. Если верхнее поле необходимо убрать, чтобы <h2> находился на одном уровне с содержимым контейнера `#nav container`, используется свойство `margin-top: 0`.

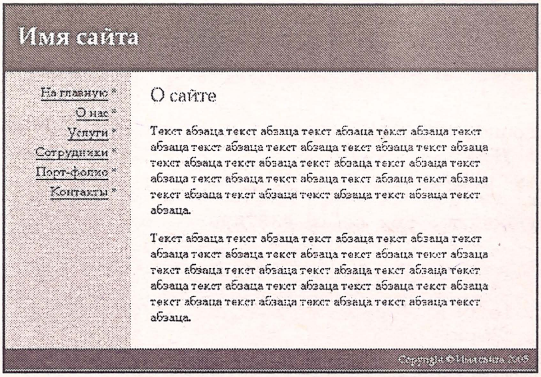
Далее необходимо изменить цвет, используя объявление `color: #B23B00`.

Стандартные заголовки принято отображать полужирным шрифтом. Данное положение по умолчанию можно изменить, используя `font-weight: normal`, как показано в листинге 19.11. Результаты выполнения данного кода представлены на рис. 19.10.

Листинг 19.11. Код CSS, определяющий стилевое оформление элемента <h2>

```
body
{
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF url(header-base.gif) repeat-y;
}
h1
{
    background: #D36832;
    color: #FFF;
    padding: 20px;
    margin: 0;
    border-bottom: 5px solid #387A9B;
}
#nav
{
    float: left;
    width: 130px;
    display: inline;
    margin-left: 20px;
    padding: 15px 0;
}
```

```
#nav ul
{
    margin: 0;
    padding: 0;
    list-style-type: none;
    text-align: right;
}
#nav li
{
    background: url(header-bullet.gif) no-repeat 100% .4em;
    padding: 0 10px 5px 0;
}
#content
{
    float: left;
    width: 475px;
    margin-left: 45px;
    padding: 15px 0;
}
#footer
{
    clear: both;
    background: #387A9B;
    color: #fff;
    padding: 5px 10px;
    text-align: right;
    font-size: 80%;
}
h2
{
    margin-top: 0;
    color: #B23B00;
    font-weight: normal;
}
```



О сайте

Текст абзаца текст абзаца.

Текст абзаца текст абзаца.

Рис. 19.10. *Элемент <h2> с переопределенным стилем*

Стилевое оформление элемента <a>

Последний этап данного урока связан с переопределением цветов ссылок. При этом мы будем работать с четырьмя псевдоклассами.

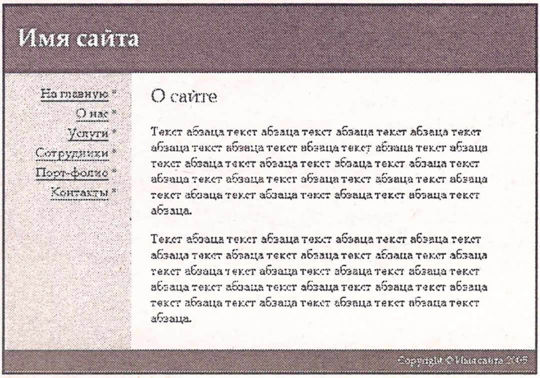
Для псевдокласса `a:link` устанавливается свойство `color: #175B7D`, для псевдокласса `a:visited` — свойство `color: #600`.

Для псевдоклассов `a:hover` и `a:active` устанавливаются соответственно свойства `color: #fff;` и `background: #175B7D`, как показано в листинге 19.12. Результаты выполнения данного кода представлены на рис. 19.11.

Листинг 19.12. Код CSS, определяющий стилевое оформление элемента `<h2>`

```
body
{
    text-align: center;
    background: #B0BFC2;
    color: #444;
}
#container
{
    text-align: left;
    margin: 0 auto;
    width: 700px;
    background: #FFF url(header-base.gif) repeat-y;
}
h1
{
    background: #D36832;
    color: #FFF;
    padding: 20px;
    margin: 0;
    border-bottom: 5px solid #387A9B;
}
#nav
{
    float: left;
    width: 130px;
    display: inline;
    margin-left: 20px;
    padding: 15px 0;
}
```

```
#nav ul
{
    margin: 0;
    padding: 0;
    list-style-type: none;
    text-align: right;
}
#nav li
{
    background: url(header-bullet.gif) no-repeat 100% .4em;
    padding: 0 10px 5px 0;
}
#content
{
    float: left;
    width: 475px;
    margin-left: 45px;
    padding: 15px 0;
}
#footer
{
    clear: both;
    background: #387A9B;
    color: #fff;
    padding: 5px 10px;
    text-align: right;
    font-size: 80%;
}
h2
{
    margin-top: 0;
    color: #B23B00;
    font-weight: normal;
}
a:link
{
    color: #175B7D;
}
a:visited
{
    color: #600;
}
a:hover, a:active
{
    color: #fff;
    background: #175B7D;
}
```



О сайте

Текст абзаца текст абзаца.

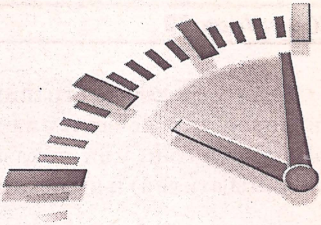
Текст абзаца текст абзаца.

Рис. 19.11. Элемент <a> с переопределенным стилем

Резюме

Из этого урока вы узнали, как выровнять по центру содержимое окна браузера, определить стиль элемента `<h1>` и разместить на странице две колонки и примечание. В следующем уроке вы научитесь, используя CSS, определять стиль страницы, предназначенной для печати.

Урок 20



Подготовка страницы для печати

В этом уроке вы научитесь, используя CSS, создавать стиль страницы для печати. Кроме того, вы узнаете о типах среды и их использовании.

Настройки CSS для печати

В прошлом многие сайты предлагали две версии каждой страницы — для отображения на экране и для печати. В общем случае содержимое двух страниц не отличалось, оно просто представлялось различными способами.

Одно из достоинств CSS заключается в том, что можно разработать стиль страницы для печати, не создавая при этом дополнительных страниц. Для этого используется атрибут `media`, который можно применять к ссылкам на внешние таблицы стилей (как показано в листинге 20.1) или к элементу `style` внедренных стилей (как показано в листинге 20.2).

Листинг 20.1. Код HTML, содержащий ссылку на внешнюю таблицу стилей и значение `media`, равное `screen`

```
<link rel="stylesheet" href="lesson20.css" type="text/css"
media="screen">
```

Листинг 20.2. Код HTML, содержащий внедренный стиль и значение `media`, равное `screen`

```
<style type="text/css" media="screen">
<!--
. Screen style rules go here ...
-->
</style>
```

Если структура страницы должна быть одинаковой для экрана и для печати, значение атрибута `media` необходимо установить так, как показано в листинге 20.3 (для внешней таблицы стилей) и в листинге 20.4 (для внедренных стилей).

Листинг 20.3. Код HTML, содержащий ссылку на внешнюю таблицу стилей и значение `media`, равное `all`

```
<link rel="stylesheet" href="lesson20.css" type="text/css"
media="all">
```

Листинг 20.4. Код HTML, содержащий внедренную таблицу стилей и значение `media`, равное `all`

```
<style type="text/css" media="all">
<!--
. All style rules go here ...
-->
</style>
```

Если структура страницы должна отличаться при отображении на экране и при печати, значения атрибута `media` должны устанавливаться равными `screen` и `print`, как показано в листингах 20.5 и 20.6.

Листинг 20.5. Код HTML, содержащий ссылки на внешние таблицы стилей и значения `media`, равные `screen` и `print`

```
<link rel="stylesheet" href="lesson20.css" type="text/css"
media="screen">
<link rel="stylesheet" href="lesson20-print.css"
type="text/css" media="print">
```

Листинг 20.6. Код HTML, содержащий внедренные таблицы стилей и значения `media`, равные `screen` и `print`

```
<style type="text/css" media="screen">
<!--
. Screen style rules go here ...
-->
</style>
<style type="text/css" media="print">
<!--
. Print style rules go here ...
-->
</style>
```

Типы среды и старые браузеры

В настоящее время в спецификации CSS существует 10 типов среды. Перечислим их:

- `all` — для всех устройств;
- `aural` — для синтезаторов речи;
- `braille` — для тактильных устройств, использующих систему Брайля;
- `embossed` — для страничных принтеров, использующих систему Брайля;
- `handheld` — для ручных устройств (например, мобильных телефонов и КПК);
- `print` — для печатных материалов и документов, отображаемых в режиме предварительного просмотра;
- `projection` — для проектируемых презентаций;
- `screen` — для цветных компьютерных мониторов;
- `tty` — для сред, использующих моноширинную сетку символов (например, телетайпы и терминалы);
- `tv` — для устройств телевизионного типа.

В одном атрибуте можно указывать несколько типов среды, разделяя их запятыми. Например, можно использовать объявление `media="print, projection"`, касающееся только указанных типов среды.

Если тип среды не указан, по умолчанию используется `screen`.

Некоторые старые браузеры, например Netscape Navigator 4, не понимают некоторые типы сред или типы, разделенные запятыми. Если вам необходима поддержка этих браузеров, лучше всего использовать значение `screen` или вообще не указывать тип среды.

Начальные коды HTML и CSS

Как показано в листинге 20.7, в данном уроке мы будем использовать код HTML из урока 19, “Расположение двух колонок с заголовком и примечанием”, и подготовим его для печати.

Листинг 20.7. Код HTML, содержащий разметку страницы с двумя колонками

```
<div id="container">
    <h1>
        Sitename
    </h1>
    <div id="nav">
        <ul>
          <li><a href="#">Home</a></li>
          <li><a href="#">About us</a></li>
          <li><a href="#">Services</a></li>
          <li><a href="#">Staff</a></li>
          <li><a href="#">Portfolio</a></li>
          <li><a href="#">Contact us</a></li>
        </ul>
    </div>
    <div id="content">
        <h2>
          About Sitename
        </h2>
        <p>
          Lorem ipsum dolor....
        </p>
        <p>
          Ut wisi enim ad...
        </p>
    </div>
    <div id="footer">
        Copyright &copy; Sitename 2005
    </div>
</div>
```

Создание селекторов для стиля для печати

Для определения стиля страницы с двумя колонками будем использовать пять селекторов, показанных в листинге 20.8.

Листинг 20.8. Код CSS, иллюстрирующий селекторы, применяемые для создания стиля для печати страницы с двумя колонками

```
body {...}
h1 {...}
#nav {...}
#footer {...}
a {...}
```

Некоторые замечания о стиле для печати

Разрабатывая стили для печати, следует помнить, что "плавающие" контейнеры и сложное абсолютное позиционирование могут в некоторых браузерах стать источниками проблем.

Длинные "плавающие" контейнеры могут вызвать проблемы в некоторых версиях Mozilla и Netscape. Эти браузеры иногда неправильно рассчитывают длину "плавающего" контейнера и печатают только первую страницу содержимого контейнера.

Сложное абсолютное позиционирование (и структуры, которые должны идеально выравниваться по пикселям) могут вызвать проблемы в Internet Explorer 6. Этот браузер завершает работу отказом при выборе предварительного просмотра страницы с абсолютным позиционированием уровней.

Хотя некоторые "плавающие" элементы и содержимое с абсолютным позиционированием использовать можно, при подготовке к печати общую структуру лучше всего сделать максимально простой.

Многие браузеры по умолчанию не печатают фоновые изображения. Поэтому эти изображения не должны содержать информации, критичной для читателя.

Стилевое оформление элемента `<body>`

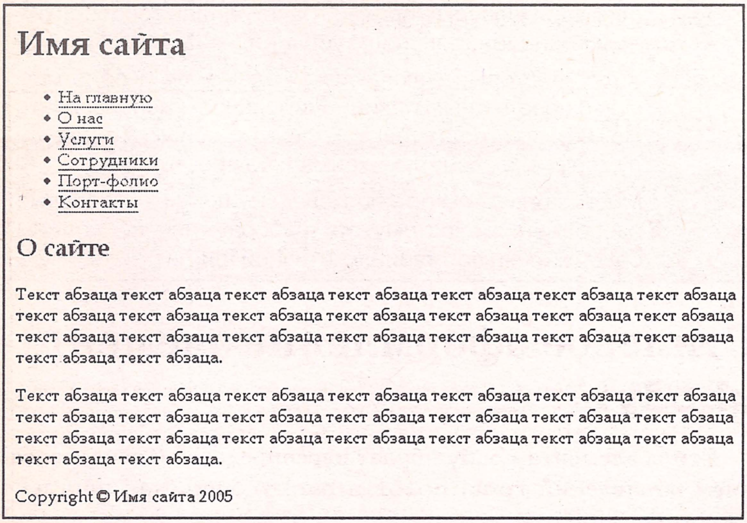
Стиль элемента `<body>` будет переопределен с использованием объявлений `font`, `color` и `background`, как показано в листинге 20.9.

Свойство `font` используется для определения стандартного размера шрифта и семейства шрифтов для всех элементов страницы. В данном примере используется значение `100%`, поскольку это позволяет пользователю контролировать размер шрифта. Кроме того, выбран шрифт с засечками (`serif`), поскольку он является более читабельным на печати, тогда как шрифт без засечек (`sans-serif`) удобнее на экране или мониторе (например, на небольших экранах мобильных телефонов шрифты с засечками читать практически невозможно). Все настройки вы можете изменить по своему усмотрению.

Для элемента `<body>` не задаются ни поля, ни заполнение; все соответствующие настройки определяются принтером. Результаты выполнения данного кода представлены на рис. 20.1.

Листинг 20.9. Код CSS, определяющий стилевое оформление тела

```
body
{
    font: 100% georgia, times, serif;
    background: #fff;
    color: #000;
}
```



Имя сайта

- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

О сайте

Текст абзаца текст абзаца.

Текст абзаца текст абзаца.

Copyright © Имя сайта 2005

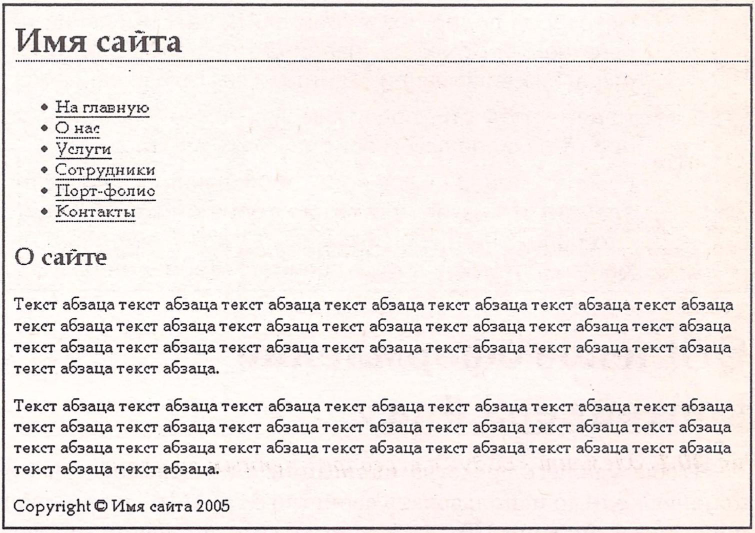
Рис. 20.1. Элемент <body> с переопределенным стилем

Стилевое оформление элемента <h1>

Стиль элемента `<h1>` будет очень простым; для него задаются только свойства `border-bottom` и `margin-bottom`, как показано в листинге 20.10 и на рис. 20.2.

Листинг 20.10. Код CSS, определяющий стилевое оформление элемента `<h1>`

```
body
{
    font: 100% georgia, times, serif;
    background: #fff;
    color: #000;
}
h1
{
    border-bottom: 1px solid #999;
    margin-bottom: 1em;
}
```



Имя сайта

- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

О сайте

Текст абзаца текст абзаца.

Текст абзаца текст абзаца.

Copyright © Имя сайта 2005

Рис. 20.2. Элемент <h1> с переопределенным стилем

Что такое “em”

В традиционной типографии термином “em” обозначается ширина прописной буквы “M” при текущих гарнитуре шрифта и размере кегля. Ширину 1 em имеет длинное тире (“em dash”); ширину 1 en, составляющую половину ширины прописной “M”, — короткое тире (“en dash”).

В CSS “em” — это относительная мера длины, наследующая информацию о размере от родительских элементов. Если родительским элементом является `<body>`, размер элемента в действительности определяется настройками шрифтов в браузере пользователя. Таким образом, в большинстве браузеров, где размер шрифта равен 16 пикселям, `1em` равно `16px`.

Сокрытие содержимого от принтера

Некоторое содержимое Web-сайта бессмысленно на печатаемой странице, например информация, относящаяся к внутренней навигации сайта, или реклама.

Данные области содержимого можно спрятать от принтера, используя свойство `display: none;`.

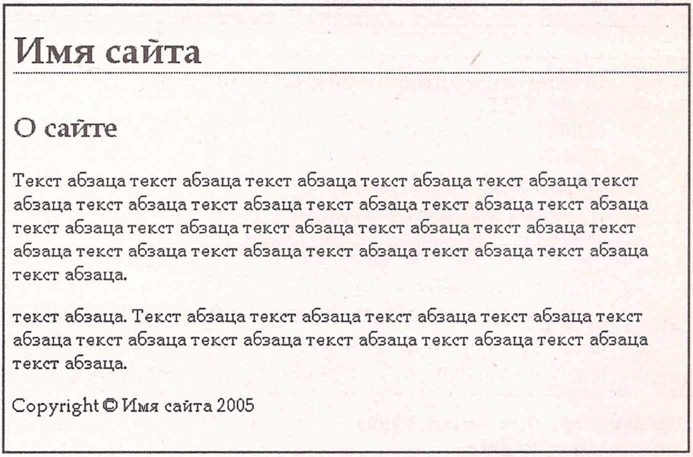
При применении данного объявления к элементу элемент и все его потомки отображаться не будут. Это положение нельзя отменить, установив для элементов-потомков другое свойство отображения.

Стилевое оформление контейнера #nav

Чтобы удалить средства навигации с печатаемой страницы, необходимо использовать свойство `display: none;`, как показано в листинге 20.11. Результаты выполнения данного кода представлены на рис. 20.3.

Листинг 20.11. Код CSS, определяющий стилевое оформление контейнера `#nav`

```
body
{
    font: 100% georgia, times, serif;
    background: #fff;
    color: #000;
}
h1
{
    border-bottom: 1px solid #999;
    margin-bottom: 1em;
}
#nav
{
    display: none;
}
```



Имя сайта

О сайте

Текст абзаца текст абзаца.

текст абзаца. Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Copyright © Имя сайта 2005

Рис. 20.3. Элемент #nav с переопределенным стилем

Стилевое оформление контейнера `#footer`

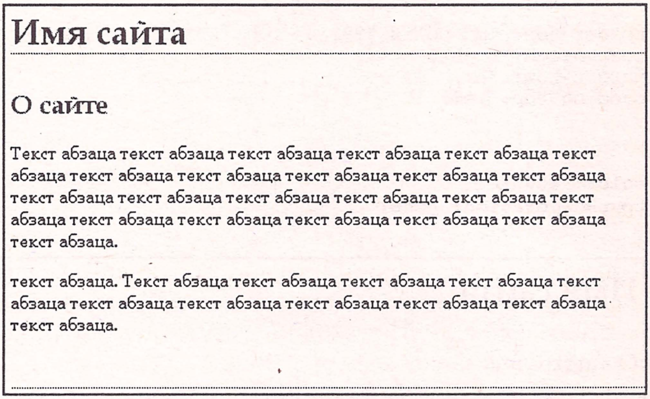
Примечание можно отделить от остального содержимого, используя свойства `border-top` и `margin-top`.

Содержимое примечания необходимо выровнять по правому краю, используя свойство `text-align`.

Чтобы добавить промежуток между границей и содержимым примечания, можно использовать свойство `padding-top`, как показано в листинге 20.12. Результаты выполнения данного кода представлены на рис. 20.4.

Листинг 20.12. Код CSS, определяющий стилевое оформление контейнера `#footer`

```
body
{
    font: 100% georgia, times, serif;
    background: #fff;
    color: #000;
}
h1
{
    border-bottom: 1px solid #999;
    margin-bottom: 1em;
}
#nav
{
    display: none;
}
#footer
{
    border-top: 1px solid #999;
    text-align: right;
    margin-top: 3em;
    padding-top: 1em;
}
```



Имя сайта

О сайте

Текст абзаца текст абзаца.

текст абзаца. Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Рис. 20.4. Элемент `#footer` с переопределенным стилем

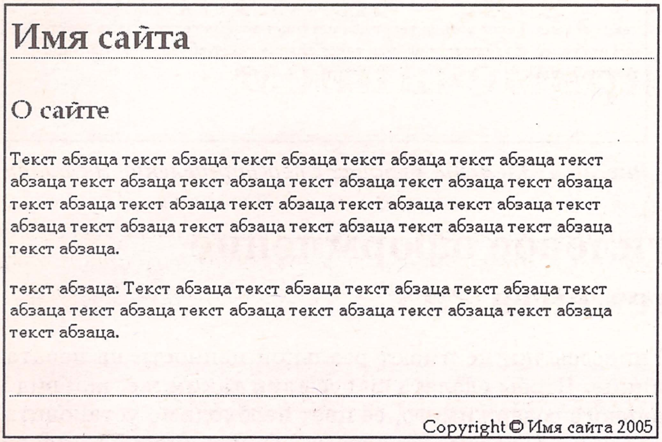
Стилевое оформление элемента `<a>`

Гиперссылки не имеют реальной ценности на печатаемой странице. Чтобы сделать вид ссылки таким же, как вид всего остального содержимого, ее цвет необходимо установить равным `#000`, а затем отключить подчеркивание, используя `text-decoration: none;`, как показано в листинге 20.13. Окончательный результат представлен на рис. 20.5.

Листинг 20.13. Код CSS, определяющий стилевое оформление элемента <a>

```
body
{
    font: 100% georgia, times, serif;
    background: #fff;
    color: #000;
}
h1
{
    border-bottom: 1px solid #999;
    margin-bottom: 1em;
}
#nav
{
    display: none;
}
#footer
```

```
{
    border-top: 1px solid #999;
    text-align: right;
    margin-top: 3em;
    padding-top: 1em;
}
a
{
    color: #000;
    text-decoration: none;
}
```



Имя сайта

О сайте

Текст абзаца текст абзаца.

текст абзаца. Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

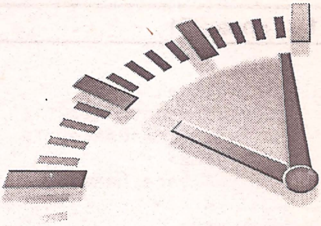
Copyright © Имя сайта 2005

Рис. 20.5. Элемент <a> с переопределенным стилем

Резюме

Из этого урока вы узнали, как, используя атрибуты среды, определять стиль материала, подготавливаемого для печати. В следующем уроке вы научитесь размещать на странице три колонки переменного размера с заголовком и примечанием.

Урок 21



Размещение трех колонок с заголовком и примечанием

В этом уроке вы научитесь размещать на странице три колонки текста с заголовком и примечанием. Используемый метод предусматривает расположение фоновых изображений внутри двух контейнеров, позволяющее создать иллюзию цветных колонок, а также использование трех “плавающих” колонок.

Подготовка HTML-кода

Хотя для формирования структуры HTML-страниц можно использовать таблицы, такое решение не идеально. С добавлением таблиц страницы существенно увеличиваются в размерах из-за необходимости дополнительной разметки. Кроме того, они становятся менее гибкими, поэтому разделы документа уже нельзя легко перемещать, не затрагивая всю разметку.

Формируя структуру с помощью CSS, мы в общем случае получаем меньшие и более гибкие документы.

HTML-код этого урока состоит из семи основных контейнеров — элемента `<h1>` и шести элементов `<div>`, как показано в листинге 21.1.

Листинг 21.1. Код HTML, содержащий разметку страницы с тремя колонками

```
<h1>
    Sitename
</h1>
```

```
<div id="container">
    <div id="container2">
        <div id="content">
          <h2>
            Page heading
          </h2>
          <p>
            Lorem ipsum dolor sit amet...
          </p>
        </div>
        <div id="news">
          <h3>
            News
          </h3>
          <p>
            Lorem ipsum dolor sit amet...
          </p>
        </div>
        <div id="nav">
          <h3>
            Sections
          </h3>
          <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">About us</a></li>
            <li><a href="#">Services</a></li>
            <li><a href="#">Staff</a></li>
            <li><a href="#">Portfolio</a></li>
            <li><a href="#">Contact us</a></li>
          </ul>
        </div>
        <div id="footer">
          Copyright &copy; Sitename 2005
        </div>
    </div>
</div>
```

Создание селекторов для определения стиля страницы с тремя колонками

Чтобы оформить стиль страницы с тремя колонками, будут использованы 10 селекторов, как показано в листинге 21.2.

Листинг 21.2. Код CSS, иллюстрирующий селекторы, которые применяются для стилевого оформления страницы с тремя колонками

```
body {...}
h1 {...}
h2, h3, p {...}
#container {...}
#container2 {...}
#content {...}
#news {...}
#nav {...}
#nav ul {...}
#footer {...}
```

Что такое "жидкая" структура?

Существует четыре общих подхода к структуре Web-страниц. Перечислим их.

- **"Жидкая" структура.** Ширина всех контейнеров на странице определена в процентах. "Жидкая" структура изменяет размер при изменении окна браузера.
- **Комбинация "жидкой" и фиксированной структуры.** Подобна "жидкой" структуре за исключением одного-двух контейнеров, имеющих фиксированную ширину.
- **Фиксированная структура.** Ширина всех контейнеров страницы задана в пикселях или других фиксированных единицах. При изменении размеров окна браузера фиксированная структура не меняется.
- **Динамическая структура, основанная на em.** Ширина всех контейнеров на странице выражена в единицах "em". Контейнеры масштабируются согласно текущему размеру шрифта в браузере пользователя. Подобные структуры не будут изменяться при изменении размера окна браузера.

Кроме того, допускаются любые комбинации перечисленных элементов.

Создание сетки “жидкой” структуры

При создании “жидкой” структуры важно создать вертикальные разделительные полосы, чтобы колонки с содержимым не перекрывались. Все размеры ширины необходимо задавать в процентах, чтобы при изменении размера в зависимости от состояния окна браузера страница представляла собой единый объект.

Ширина в процентах вычисляется браузером, поэтому при вычислениях неизбежно округление в большую или меньшую сторону. Поэтому суммарная ширина всех элементов должна быть меньше ширины окна, чтобы за счет разницы можно было сгладить возможные ошибки округления.

В нашем случае общая ширина контейнеров и их заполнения составляет 100 %. Тем не менее заполнение с правой стороны элемента `#nav` не включено в расчеты, поэтому в общей структуре остается неопределенным 3 % места. Все выбранные размеры показаны на рис 21.1.

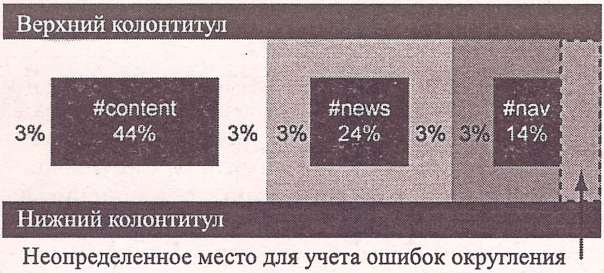


Рис. 21.1. Диаграмма сетки “жидкой” структуры

Создание фоновых изображений

В данной структуре используется два фоновых изображения, создающих иллюзию цветных колонок. Изображения должны быть достаточно широкими, чтобы заполнить наибольший экран. В данном случае их ширина выбрана равной `2000px`.

С помощью первого изображения создается цвет фона колонок `#content` и `#news`.

С помощью второго изображения создается цвет фона колонки `#nav`. Поскольку данное изображение накладывается поверх первого, оно должно иметь прозрачный фон, как показано на рис 21.2.

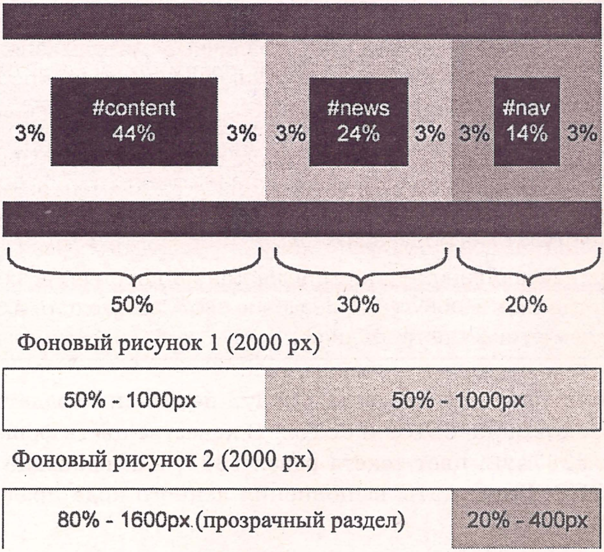

3%
#content
44%
3%
3%
#news
24%
3%
3%
#nav
14%
3%
50%
30%
20%
Фоновый рисунок 1 (2000 px)
50% - 1000px
50% - 1000px
Фоновый рисунок 2 (2000 px)
80% - 1600px (прозрачный раздел)
20% - 400px

Рис. 21.2. Фоновые изображения

Стилевое оформление элемента <body>

Завершив работу с сеткой и изображениями, можно начинать кодирование страницы.

Первый этап заключается в выравнивании содержимого страницы по границам окна браузера. Для этого свойствам `margin` и `padding` необходимо присвоить значение 0.

Для всех элементов страницы можно установить свойства `font-size` и `font-family`, задав их для элемента `<body>` с помощью объявления `font: 90% arial, helvetica, sans-serif;`.

Установка свойств `margin` и `padding` элемента `<body>`

Браузеры используют различные методы для установки отступов по умолчанию для элемента `<body>`.

Если используется объявление `padding: 0`, Opera выровняет содержимое по границам окна браузера.

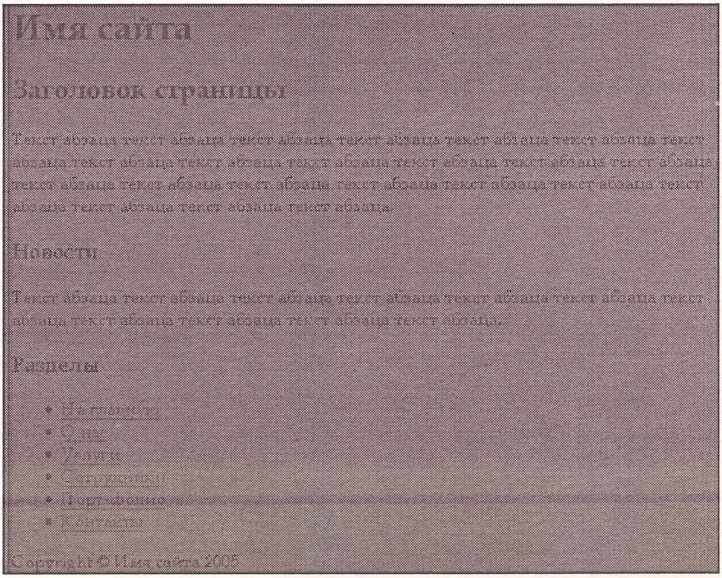
Если используется объявление `margin: 0`, все остальные браузеры выровняют содержимое по границам окна браузера.

Единственный способ унификации действий всех браузеров — обнулить и значение свойства `margin`, и значение свойства `padding`.

Кроме того, для элемента `<body>` необходимо задать свойства `background-color` и `color`. В качестве цвета фона выбирается `#387A9B`, цвет текста равен `#333`, как показано в листинге 21.3. Результаты выполнения данного кода представлены на рис. 21.3.

Листинг 21.3. Код CSS, определяющий стилевое оформление элемента `<body>`

```
body
{
    margin: 0;
    padding: 0;
    font: 90% arial, helvetica, sans-serif;
    background: #387A9B;
    color: #333;
}
```



Имя сайта

Заголовок страницы

Текст абзаца текст абзаца.

Новости

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Разделы

- На главную
- О нас
- Услуги
- Сотрудники
- Портфолио
- Контакты

Copyright © Имя сайта 2005

Рис. 21.3. Элемент <body> с переопределенным стилем

Стилевое оформление элемента <h1>

Используем элемент <h1> для создания “шапки” вверху страницы.

Вначале необходимо задать свойства `background-color` и `color`. В данном примере используем цвет фона `#D36832` и цвет текста `#fff`.

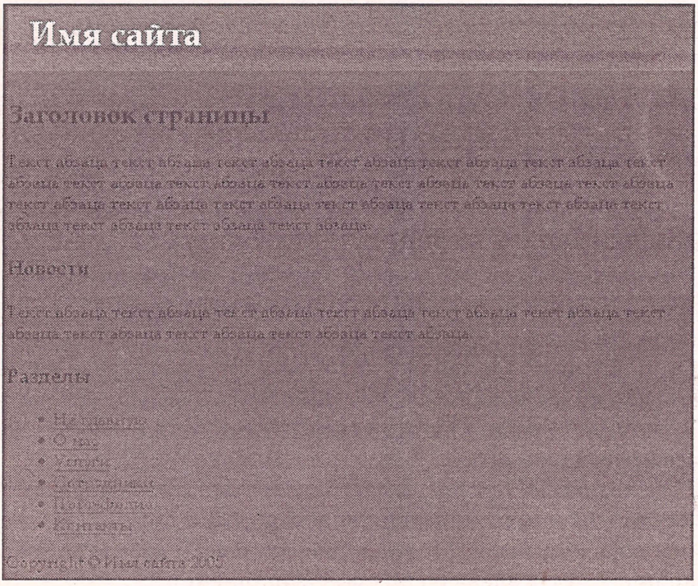
Стандартные элементы <h1> имеют предопределенные верхние и нижние поля. Чтобы поместить элемент <h1> в левый верхний угол окна браузера, этим полям необходимо присвоить значение 0.

Для создания пустого пространства вокруг содержимого элемента <h1> применяется свойство `padding: .5em 3%;`. Таким образом, вверху и внизу содержимого добавляется заполнение `.5em`, а слева и справа — заполнение `3%`.

Наконец с помощью объявления `border-bottom: 5px solid #387A9B;` вверху элемента добавляется граница, как показано в листинге 21.4 и на рис. 21.4.

Листинг 21.4. Код CSS, определяющий стилевое оформление элемента `<h1>`

```
body
{
    margin: 0;
    padding: 0;
    font: 90% arial, helvetica, sans-serif;
    background: #387A9B;
    color: #333;
}
h1
{
    background: #D36832;
    color: #FFF;
    margin: 0;
    padding: .5em 3%;
    border-bottom: 5px solid #387A9B;
}
```



Имя сайта

Заголовок страницы

Текст абзаца текст абзаца.

Новости

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Разделы

- На главную
- О нас
- Услуги
- [illegible]
- Портфолио
- Контакты

Copyright © Имя сайта 2005

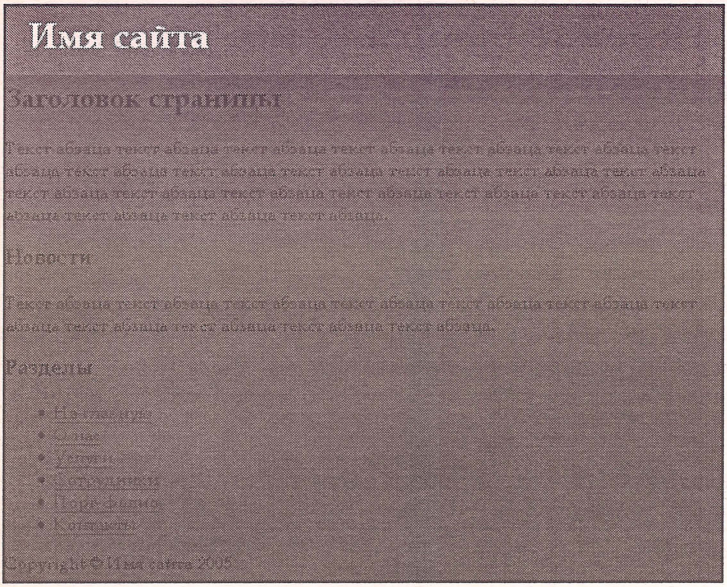
Рис. 21.4. Элемент <h1> с переопределенным стилем

Стилевое оформление элементов <h2> и <h3>

Элементы `<h2>` и `<h3>` располагаются внутри элементов `#container` и `#container2`. Одни браузеры отобразят эти элементы заголовков и их верхние границы внутри контейнера. Другие покажут только заголовки и позволят полям выходить за верхнюю границу контейнера. Подробнее этот момент объясняется во врезке “Поля в ловушке” в уроке 13, “Стилевое оформление блока цитаты”. Существует множество способов решения данной проблемы. В этом уроке верхние поля элементов `<h2>` и `<h3>` просто удаляются с помощью `margin-top: 0` (листинг 21.5 и рис. 21.5).

Листинг 21.5. Код CSS, определяющий стилевое оформление элементов `<h2>` и `<h3>`

```
body
{
    margin: 0;
    padding: 0;
    font: 90% arial, helvetica, sans-serif;
    background: #387A9B;
    color: #333;
}
h1
{
    background: #D36832;
    color: #FFF;
    margin: 0;
    padding: .5em 3%;
    border-bottom: 5px solid #387A9B;
}
h2, h3
{
    margin-top: 0;
}
```



Имя сайта

Заголовок страницы

Текст абзаца текст абзаца.

Новости

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Разделы

- На главную
- О нас
- Услуги
- Сотрудники
- Портфолио
- Контакты

Copyright © Имя сайта 2005

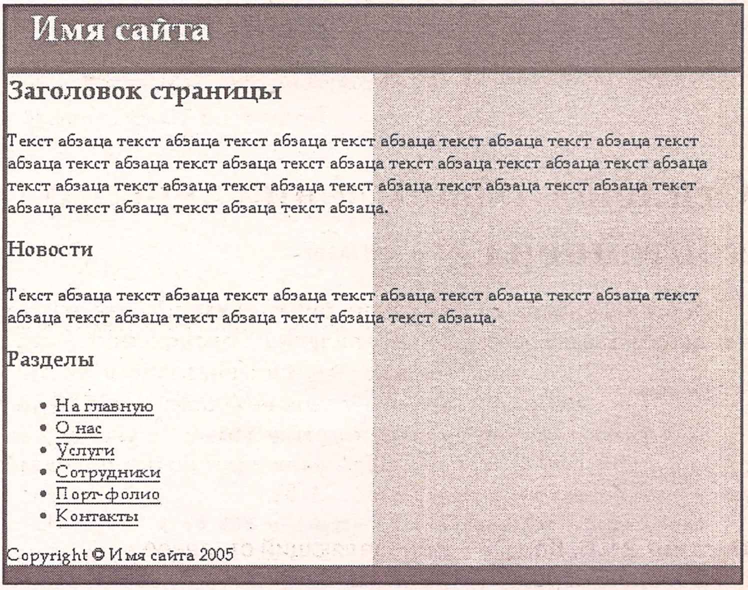
Рис. 21.5. Элементы <h2> и <h3> с переопределенным стилем

Стилевое оформление первого контейнера

Первое фоновое изображение будет связано с элементом `#container` посредством объявления `background: url (back01.gif) repeat-y 50% 0;`, как показано в листинге 21.6. Таким образом, это изображение (`back01.gif`) будет повторяться сверху вниз по оси *y*. Изображение размещается так, чтобы 50 % его площади занимали точно половину фона элемента `#container` (рис. 21.6).

Листинг 21.6. Код CSS, определяющий стилевое оформление первого контейнера

```
body
{
    margin: 0;
    padding: 0;
    font: 90% arial, helvetica, sans-serif;
    background: #387A9B;
    color: #333;
}
h1
{
    background: #D36832;
    color: #FFF;
    margin: 0;
    padding: .5em 3%;
    border-bottom: 5px solid #387A9B;
}
h2, h3
{
    margin-top: 0;
}
#container
{
    background: url(back01.gif) repeat-y 50% 0;
}
```



Имя сайта

Заголовок страницы

Текст абзаца текст абзаца.

Новости

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Разделы

- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

Copyright © Имя сайта 2005

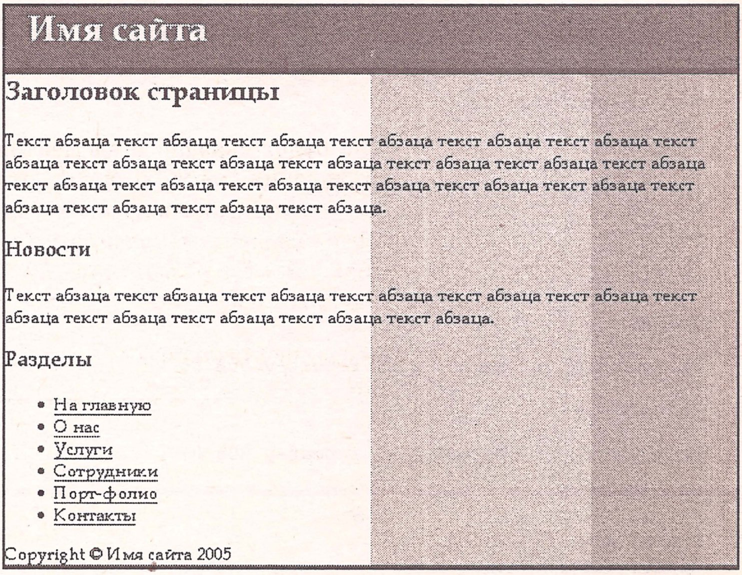
Рис. 21.6. Первый контейнер с переопределенным стилем

Стилевое оформление второго контейнера

Второе фоновое изображение будет связано с элементом `#container2` посредством объявления `background: url (back02.gif) repeat-y 80% 0;`, как показано в листинге 21.7 и на рис. 21.7. Подобно предыдущим правилам элемента `#container`, 80 % изображения будут занимать 80 % окна браузера и повторяться вниз по оси *y*.

Листинг 21.7. Код CSS, определяющий стилевое оформление второго контейнера

```
body
{
    margin: 0;
    padding: 0;
    font: 90% arial, helvetica, sans-serif;
    background: #387A9B;
    color: #333;
}
h1
{
    background: #D36832;
    color: #FFF;
    margin: 0;
    padding: .5em 3%;
    border-bottom: 5px solid #387A9B;
}
h2, h3
{
    margin-top: 0;
}
#container
{
    background: url(back01.gif) repeat-y 50% 0;
}
#container2
{
    background: url(back02.gif) repeat-y 80% 0; }
}
```



Имя сайта

Заголовок страницы

Текст абзаца текст абзаца.

Новости

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Разделы

- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

Copyright © Имя сайта 2005

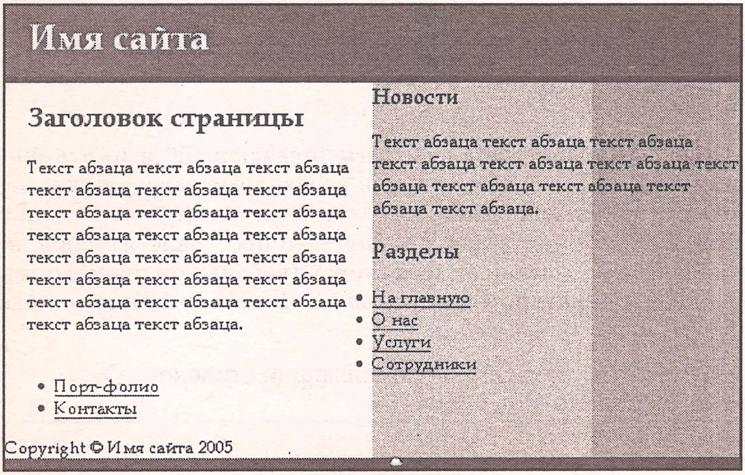
Рис. 21.7. Второй контейнер с переопределенным стилем

Стилевое оформление колонки `#content`

Первая колонка должна быть “плавающей” и иметь ширину 44 %. Кроме того, для нее необходимо задать поля, используя объявление `margin: 1em 3%;`, как показано в листинге 21.8. Мы получаем 1 em пустого пространства над контейнером и 3 % полей со всех сторон (небольшой промежуток между колонками). Результаты выполнения данного кода представлены на рис. 21.8.

Листинг 21.8. Код CSS, определяющий стилевое оформление столбца `#content`

```
body
{
    margin: 0;
    padding: 0;
    font: 90% arial, helvetica, sans-serif;
    background: #387A9B;
    color: #333;
}
h1
{
    background: #D36832;
    color: #FFF;
    margin: 0;
    padding: .5em 3%;
    border-bottom: 5px solid #387A9B;
}
h2, h3
{
    margin-top: 0;
}
#container
{
    background: url(back01.gif) repeat-y 50% 0;
}
#container2
{
    background: url(back02.gif) repeat-y 80% 0;
}
#content
{
    width: 44%;
    float: left;
    margin: 1em 3%;
}
```



Имя сайта

Заголовок страницы

Текст абзаца текст абзаца.

- Порт-фолио
- Контакты

Copyright © Имя сайта 2005

Новости

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Разделы

- На главную
- О нас
- Услуги
- Сотрудники

Рис. 21.8. Колонка `#content` *с переопределенным стилем*

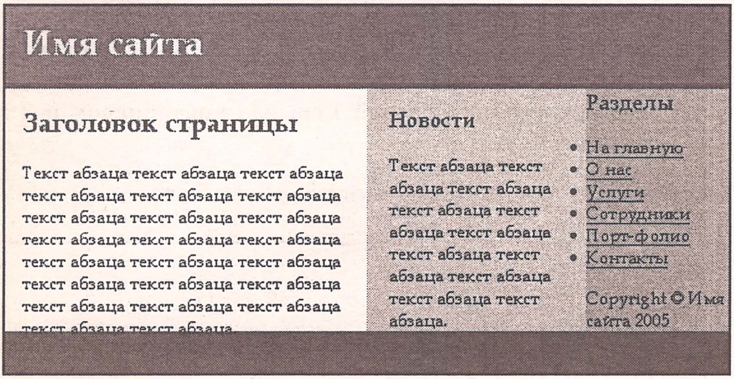
Стилевое оформление колонки #news

Вторая колонка должна быть “плавающей” и иметь ширину 24 %. Как и первая, она должна иметь поля `1 em 3 %;`, как показано в листинге 21.9 и на рис. 21.9. Как и в случае с колонкой `#content`, выполнение приведенного кода даст пустое пространство величиной 1 em над контейнером и 3 % поля с обеих сторон.

Листинг 21.9. Код CSS, определяющий стилевое оформление столбца #news

```
body
{
    margin: 0;
    padding: 0;
    font: 90% arial, helvetica, sans-serif;
    background: #387A9B;
    color: #333;
}
h1
{
    background: #D36832;
    color: #FFF;
    margin: 0;
    padding: .5em 3%;
    border-bottom: 5px solid #387A9B;
}
h2, h3
{
    margin-top: 0;
}
#container
{
    background: url(back01.gif) repeat-y 50% 0;
}
#container2
{
    background: url(back02.gif) repeat-y 80% 0;
}
#content
{
    width: 44%;
    float: left;
    margin: 1em 3%;
}
```

```
#news
{
    width: 24%;
    float: left;
    margin: 1em 3%;
}
```



Имя сайта

Заголовок страницы

Текст абзаца текст абзаца.

Новости

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Разделы

- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

Copyright © Имя сайта 2005

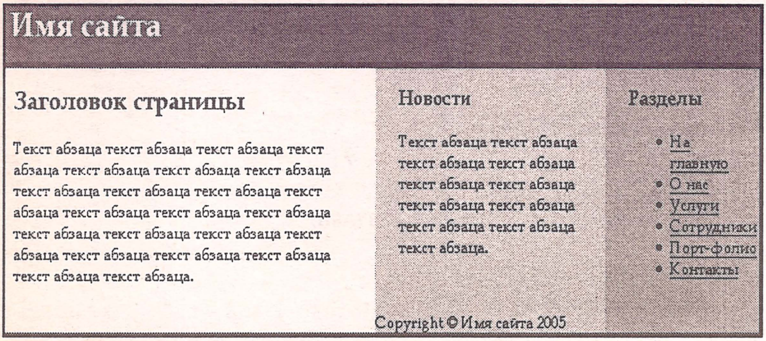
Рис. 21.9. Колонка `#news` с переопределенным стилем

Стилевое оформление колонки #nav

Третья колона должна быть “плавающей” и иметь ширину 14%. Подобно другим колонкам, ей также требуется заполнение. Однако заполнение для этого элемента задается с помощью свойства `padding: 1em 0 1em 3%;`, как показано в листинге 21.10. Заполнение по правому краю не определено, поскольку данный промежуток не определяется с целью учета ошибок округления. Результаты выполнения данного кода представлены на рис. 21.10.

Листинг 21.10. Код CSS, определяющий стилевое оформление столбца #nav

```
body
{
    margin: 0;
    padding: 0;
    font: 90% arial, helvetica, sans-serif;
    background: #387A9B;
    color: #333;
}
h1
{
    background: #D36832;
    color: #FFF;
    margin: 0;
    padding: .5em 3%;
    border-bottom: 5px solid #387A9B;
}
h2, h3
{
    margin-top: 0;
}
#container
{
    background: url(back01.gif) repeat-y 50% 0;
}
#container2
{
    background: url(back02.gif) repeat-y 80% 0;
}
#content
{
    width: 44%;
    float: left;
    margin: 1em 3%;
}
#news
{
    width: 24%;
    float: left;
    margin: 1em 3%;
}
#nav
{
    width: 14%;
    float: left;
    margin: 1em 0 1em 3%;
}
```



Имя сайта

Заголовок страницы

Текст абзаца текст абзаца.

Новости

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Разделы

- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

Copyright © Имя сайта 2005

Рис. 21.10. Колонка `#nav` *с переопределенным стилем*

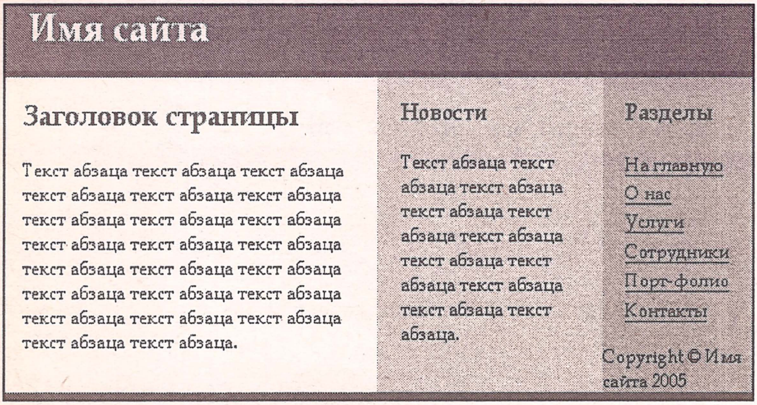
Стилевое оформление элемента `<ul>`

Третья колонка содержит навигационную панель (список). В этом уроке позиции списка будут выровнены по границе контейнера без использования маркеров. Для этого применяются три объявления — `margin: 0;`, `padding: 0;` и `list-style-type: none;`. Чтобы добавить пустое пространство между позициями списка, можно увеличить высоту строки, например, используя `line-height: 150%;`, как показано в листинге 21.11 и на рис. 21.11.

Листинг 21.11. Код CSS, определяющий стилевое оформление элемента `<ul>`

```
body
{
    margin: 0;
    padding: 0;
    font: 90% arial, helvetica, sans-serif;
    background: #387A9B;
    color: #333;
}
h1
```

```
{
    background: #D36832;
    color: #FFF;
    margin: 0;
    padding: .5em 3%;
    border-bottom: 5px solid #387A9B;
}
h2, h3
{
    margin-top: 0;
}
#container
{
    background: url(back01.gif) repeat-y 50% 0;
}
#container2
{
    background: url(back02.gif) repeat-y 80% 0;
}
#content
{
    width: 44%;
    float: left;
    margin: 1em 3%;
}
#news
{
    width: 24%;
    float: left;
    margin: 1em 3%;
}
#nav
{
    width: 14%;
    float: left;
    margin: 1em 0 1em 3%;
}
#nav ul
{
    margin: 0;
    padding: 0;
    list-style-type: none;
    line-height: 150%;
}
```



Имя сайта

Заголовок страницы

Текст абзаца текст абзаца.

Новости

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Разделы

Copyright © Имя сайта 2005

Рис. 21.11. Элемент `<ul>` *с переопределенным стилем*

Стилевое оформление элемента `#footer`

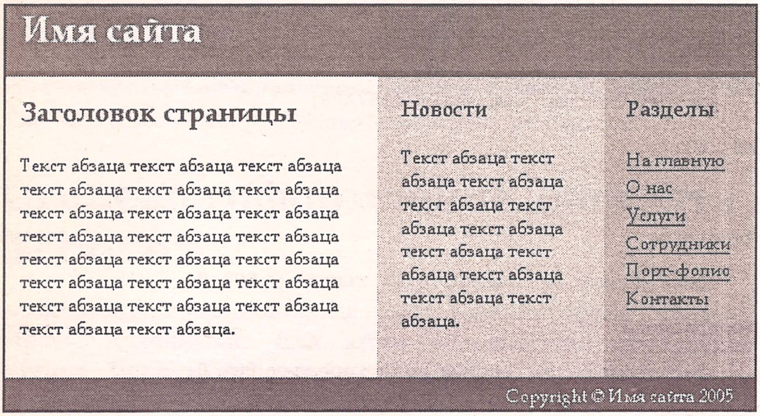
Элемент `#footer` должен находиться под тремя “плавающими” колонками. Для этого используется объявление `clear: both;`. Свойства `background-color` и `color` задаются с использованием объявлений `background: #387A9B;` и `color: #fff;`.

Далее к элементу `#footer` необходимо добавить заполнения, чтобы немного увеличить пустое пространство вокруг содержимого. Для этого используется объявление `padding: 5px 3%;`.

Наконец содержимое примечания можно выровнять по правому краю, используя объявление `text-align: right;`, как показано в листинге 21.12. Результаты выполнения данного кода представлены на рис. 21.12.

Листинг 21.12. Код CSS, определяющий стилевое оформление элемента `#footer`

```
body
{
    margin: 0;
    padding: 0;
    font: 90% arial, helvetica, sans-serif;
    background: #387A9B;
    color: #333;
}
h1
{
    background: #D36832;
    color: #FFF;
    margin: 0;
    padding: .5em 3%;
    border-bottom: 5px solid #387A9B;
}
h2, h3
{
    margin-top: 0;
}
#container
{
    background: url(back01.gif) repeat-y 50% 0;
}
#container2
{
    background: url(back02.gif) repeat-y 80% 0;
}
#content
{
    width: 44%;
    float: left;
    margin: 1em 3%;
}
#news
{
    width: 24%;
    float: left;
    margin: 1em 3%;
}
#nav
{
    width: 14%;
    float: left;
    margin: 1em 0 1em 3%;
}
#nav ul
{
    margin: 0;
    padding: 0;
    list-style-type: none;
    line-height: 150%;
}
#footer
{
    clear: both;
    background: #387A9B;
    color: #fff;
    padding: 5px 3%;
    text-align: right;
}
```



Имя сайта

Заголовок страницы

Текст абзаца текст абзаца.

Новости

Текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца.

Разделы

- На главную
- О нас
- Услуги
- Сотрудники
- Порт-фолио
- Контакты

Copyright © Имя сайта 2005

Рис. 21.12. Окончательный вид страницы

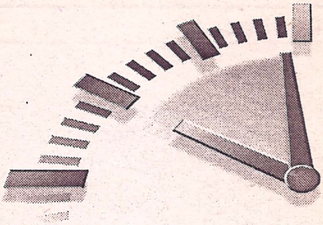
“Схлопывание” “жидких” структур

“Жидкие” структуры с тремя колонками в общем случае растягиваются и сжимаются в соответствии с шириной окна браузера. При уменьшении ширины окна один или несколько контейнеров могут опуститься ниже первой колонки страницы. Почему это происходит? Если горизонтального пространства текущей строки недостаточно для “плавающей” колонки, она опускается вниз строчка за строчкой, пока не поместится.

Резюме

Из этого урока вы узнали, как создать “жидкую” структуру с тремя колонками разного цвета. Кроме того, вы научились переопределять стиль элемента <h1>, превращая его в “шапку” вверху страницы. В следующем уроке вы научитесь исправлять основные ошибки CSS и получите несколько подсказок по улаживанию конфликтов CSS.

Урок 22



Устранение ошибок CSS

В этом уроке вы научитесь исправлять основные ошибки CSS. Кроме того, вы получите несколько подсказок по устранению ошибок CSS.

Подготовка кода CSS

Код CSS для этого урока показан в листинге 22.1. Он содержит 12 наиболее распространенных проблем CSS, которые мы будем исправлять в данном уроке.

Листинг 22.1. Код CSS, иллюстрирующий 12 наиболее распространенных ошибок

```
body
{
    font-family: times, times new roman, serif;
}
#container
{
    border: 1px gray;
    background-image: url("background.jpg");
    background-repeat: repeat-x;
    background-attach: fixed;
    width: 700;
}
h1
{
    font-size: 200%;
    color: none;
}
.introductionText
{
    font-weight: bold;
}
h2
```

```
{
    font-size: 120%
    font-weight: normal;
    color: #34a32;
}
p, ul,
{
    font-size: 80%;
    color: 333;
}
a:visited
{
    color: purple;
}
a:link
{
    color: blue;
}
a:hover
{
    color: red;
}
a:active
{
    color: black;
}
#container p
{
    color: #000;
}
p.intro
{
    color: #900;
}
```

Решение проблем

Проблема 1. В наборе правил для элемента `<body>` **имя семейства шрифтов Times New Roman содержит пробелы.** Любое название шрифта, содержащее пробелы, должно заключаться в кавычки. Проблемная строка исходного кода показана в листинге 22.2, а в листинге 22.3 приведен исправленный код.

Листинг 22.2. Код CSS, иллюстрирующий имя семейства шрифтов без кавычек

```
body
{
    font-family: times, times new roman, serif;
}
```

Листинг 22.3. Код CSS, иллюстрирующий имя семейства шрифтов в кавычках

```
body
{
    font-family: times, "times new roman", serif;
}
```

Проблема 2. Свойство `border` **в наборе правил** `#container` **не содержит заданного свойства** `border-style`. Граница отображаться не будет, поскольку по умолчанию значение `border-style` равно `none`. Поэтому необходимо задать значение `border-style` равным `dotted`, `dashed`, `solid`, `double`, `grooved`, `ridged`, `inset` или `outset`. Проблемная строка исходного кода показана в листинге 22.4, а в листинге 22.5 приведен исправленный код.

Листинг 22.4. Код CSS, иллюстрирующий неправильное объявление свойства `border`

```
#container
{
    border: 1px gray;
    background-image: url("background.jpg");
    background-repeat: repeat-x;
    background-attach: fixed;
    width: 700;
}
```

Листинг 22.5. Код CSS, иллюстрирующий правильное объявление свойства `border`

```
#container
{
    border: 1px solid gray;
    background-image: url("background.jpg");
    background-repeat: repeat-x;
    background-attach: fixed;
    width: 700;
}
```

Проблема 3. Свойство `background-attach` **не существует.** Правильным свойством является `background-attachment`. Проблемная строка исходного кода показана в листинге 22.6, а в листинге 22.7 приведен исправленный код.

Листинг 22.6. Код CSS, иллюстрирующий неверное свойство

```
#container
{
    border: 1px solid gray;
    background-image: url("background.jpg");
    background-repeat: repeat-x;
    background-attach: fixed;
    width: 700;
}
```

Листинг 22.7. Код CSS, иллюстрирующий верное свойство

```
#container
{
    border: 1px solid gray;
    background-image: url("background.jpg");
    background-repeat: repeat-x;
    background-attachment: fixed;
    width: 700;
}
```

Проблема 4. Значение `width` **в наборе правил** `#container` **не содержит единицы изменения.** Браузеры должны догадаться, в чем автор задал ширину — в пунктах, пиках, пикселях, единицах em или ex, миллиметрах, сантиметрах, дюймах или процентах. Проблемная строка исходного кода показана в листинге 22.8, а в листинге 22.9 приведен исправленный код.

Листинг 22.8. Код CSS, иллюстрирующий размер, заданный без единицы изменения

```
#container
{
    border: 1px solid gray;
    background-image: url("background.jpg");
    background-repeat: repeat-x;
    background-attachment: fixed;
    width: 700;
}
```

Листинг 22.9. Код CSS, иллюстрирующий размер, заданный в пикселях

```
#container
{
    border: 1px solid gray;
    background-image: url("background.jpg");
    background-repeat: repeat-x;
    background-attachment: fixed;
    width: 700px;
}
```

Проблема 5. В наборе правил `<h1>` код цвета задан как `none`. Использовано неприемлемое значение. Данная величина должна задаваться как шестнадцатеричный RGB-код, ключевое слово, ключевое слово пользовательского интерфейса или десятичный RGB-код. Автор может указать, чтобы код цвета наследовался от родительского элемента; в таком случае следует использовать значение `inherit`. Проблемная строка исходного кода показана в листинге 22.10, а в листинге 22.11 приведен исправленный код.

Листинг 22.10. Код CSS, иллюстрирующий неприемлемый код цвета

```
h1
{
    font-size: 200%;
    color: none;
}
```

Листинг 22.11. Код CSS, иллюстрирующий допустимый код цвета

```
h1
{
    font-size: 200%;
    color: inherit;
}
```

Проблема 6. Некоторые авторы предпочитают использовать имена, состоящие из прописных и строчных букв. Чтобы класс был использован, в коде HTML и селекторе он должен быть представлен одинаково.

Предположим, что код HTML содержит класс `IntroductionText`. В приведенном в начале урока коде автор использовал селектор `.introductionText`. Этот селектор не соответствует имени класса HTML, поэтому объявленное правило не будет к нему применимо. Проблемная строка исходного кода показана в листинге 22.12, а в листинге 22.13 приведен исправленный код.

Листинг 22.12. Код CSS, иллюстрирующий неправильное написание имени класса

```
.introductionText
{
    font-weight: bold;
}
```

Листинг 22.13. Код CSS, иллюстрирующий правильное написание имени класса

```
.IntroductionText
{
    font-weight: bold;
}
```

Проблема 7. В наборе правил `<h2>` **пропущено объявление** `font-size`. Браузеры прочтут следующее объявление, `font-weight: normal;`, как часть объявления `font-size`, и получат неправильный код. В результате оба объявления будут проигнорированы. Проблемная строка исходного кода показана в листинге 22.14, а в листинге 22.15 приведен исправленный код.

Листинг 22.14. Код CSS, иллюстрирующий объявление с пропущенной точкой с запятой

```
h2
{
    font-size: 120%
    font-weight: normal;
    color: #34a32;
}
```

Листинг 22.15. Код CSS, иллюстрирующий правильное объявление

```
h2
{
    font-size: 120%;
        font-weight: normal;
    color: #34a32;
}
```

Проблема 8. Пропущена цифра шестнадцатеричного числа в наборе правил `<h2>`**.** Шестнадцатеричные числа должны состоять из трех или шести цифр. Проблемная строка исходного кода показана в листинге 22.16, а в листинге 22.17 приведен исправленный код.

Листинг 22.16. Код CSS, иллюстрирующий неверное шестнадцатеричное число

```
h2
{
    font-size: 120%;
```

```
        font-weight: normal;
    color: #34a32;
}
```

Листинг 22.17. Код CSS, иллюстрирующий правильное шестнадцатеричное число

```
h2
{
    font-size: 120%;
        font-weight: normal;
    color: #34a323;
}
```

Проблема 9. После перечисления селекторов поставлена лишняя запятая: `p, ul,`. Из-за этого весь набор правил будет проигнорирован. Проблемная строка исходного кода показана в листинге 22.18, а в листинге 22.19 приведен исправленный код.

Листинг 22.18. Код CSS, иллюстрирующий запятую в конце селекторов

```
p, ul,
{
    font-size: 80%;
    color: 333;
}
```

Листинг 22.19. Код CSS, иллюстрирующий правильный синтаксис при использовании нескольких селекторов

```
p, ul
{
    font-size: 80%;
    color: 333;
}
```

Проблема 10. В коде цвета `#333` **отсутствует символ шестнадцатеричных значений** `#`. Хотя одни браузеры принимают подобное объявление, для других оно недопустимо. Проблемная строка исходного кода показана в листинге 22.20, а в листинге 22.21 приведен исправленный код.

Листинг 22.20. Код CSS, иллюстрирующий код цвета без символа #

```
p, ul
{
    font-size: 80%;
    color: 333;
}
```

Листинг 22.21. Код CSS, иллюстрирующий правильное объявление

```
p, ul
{
    font-size: 80%;
    color: #333;
}
```

Проблема 11. Псевдокласс `a:hover` **не применяется, поскольку он указан перед псевдоклассом** `a:link`. Объявления псевдоклассов необходимо поменять местами. Проблемная строка исходного кода показана в листинге 22.22, а в листинге 22.23 приведен исправленный код.

Листинг 22.22. Код CSS, иллюстрирующий использование псевдокласса `a:hover` перед псевдоклассом `a:link`

```
a:hover
{
    color: red;
}
a:link
{
    color: blue;
}
```

Листинг 22.23. Код CSS, иллюстрирующий правильный порядок псевдоклассов

```
a:link
{
    color: blue;
}
a:hover
{
    color: red;
}
```

Проблема 12. Набор правил `#container p { color: black; }` **задает черный цвет текста для всех абзацев внутри элемента** `#container`. Следующий набор правил, `p.intro { color: #900; }`, должен был задать цвет текста первого абзаца элемента `#container` равным `#900`.

Однако набор правил `p.intro` не будет применен, поскольку набор правил `#container p` имеет более высокий приоритет. Из-за того что селекторы, которые содержат идентификаторы, имеют более высокий приоритет, чем селекторы с классами, текст всех абзацев внутри элемента `#container` будет представлен черным цветом.

Чтобы задействовать набор правил `p.intro`, селектор необходимо изменить на `#container p.intro`, что повысит его приоритет. Проблемная строка исходного кода показана в листинге 22.24, а в листинге 22.25 приведен исправленный код.

Листинг 22.24. Код CSS, иллюстрирующий набор правил без идентификатора

```
#container p
{
    color: #000;
}
p.intro
{
    color: #900;
}
```

Листинг 22.25. Код CSS, иллюстрирующий набор правил с идентификатором

```
#container p
{
    color: #000;
}
#container p.intro
{
    color: #900;
}
```

Советы по устранению проблем CSS

Совет 1. Не забудьте верифицировать файлы HTML и CSS. Семь из двенадцати перечисленных выше проблем можно практически мгновенно решить с помощью средства верификации CSS.

Средство верификации HTML от W3C можно найти по адресу `http://validator.w3.org/`. Средство верификации CSS от W3C доступно по адресу `http://jigsaw.w3.org/css-validator/`.

Совет 2. Лучший способ избежать проблем (особенно если вы являетесь новичком в использовании CSS) — создавать документы поэтапно, проверяя результат каждого этапа на нескольких браузерах. Начните с общей структуры, расположите необходимые элементы и текст, а затем проверьте полученный результат в нескольких браузерах. Убедившись в устойчивости каркаса, можно начинать детальную разработку элементов.

Совет 3. Если на странице наблюдается локальная проблема, часто помогает включение границ, чтобы были видны элементы и их взаимосвязь. Чтобы сделать границы видимыми, можно воспользоваться кодом, приведенным в листинге 22.26.

Листинг 22.26. Код CSS, иллюстрирующий набор правил, делающих границы видимыми

```
li a { border: 1px solid red; }
```

Совет 4. Для быстрой локализации ошибок можно последовательно закомментировать наборы правил (как показано в листинге 22.27), наблюдая за получаемыми эффектами. Обнаружив проблемный набор правил, можно последовательно закомментировать все его объявления, пока не будет найдено правило с ошибкой.

Листинг 22.27. Код CSS, иллюстрирующий отключение набора правил с помощью комментария

```
h2
{
    font-size: 120%;
    font-weight: normal;
    color: #34a323;
}
/*
p, ul,
{
    font-size: 80%;
    color: #333;
}
*/
```

Совет 5. Если это возможно, используйте в начале документа (X)HTML полное объявление типа документа. Все документы (X)HTML обязаны содержать объявление типа документа, в котором указывается версия (X)HTML, использованная в документе. С помощью данной информации Web-браузеры определяют, какой режим визуализации использовать. Если в документе присутствует правильное и полное объявление типа документа, многие Web-браузеры переключаются в стандартный режим, т.е. более тесно придерживаются спецификаций CSS. Основные типы документов иллюстрируются в листингах 22.28–22.32.

Листинг 22.28. Код HTML, демонстрирующий объявление строгого типа документа HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Листинг 22.29. Код HTML, демонстрирующий объявление переходного типа документа HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Листинг 22.30. Код HTML, демонстрирующий объявление строгого типа документа XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Листинг 22.31. Код HTML, демонстрирующий объявление переходного типа документа XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Листинг 22.32. Код HTML, демонстрирующий объявление типа документа XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Резюме

Из этого урока вы узнали, как исправлять характерные ошибки CSS, и получили несколько советов, помогающих избежать таких ошибок.

Предметный указатель

C

CSS
  блочная модель, 51
  использование, 19
  использование правил, 21
  настройка для печати, 205
  советы по устранению проблем, 249
  характерные ошибки, 241

H

HTML-код
  подготовка, 67

А

Абзац, 111

Б

Блок
  ширина, 52
Блок объявления, 22

В

Визуальное выделение, 146
Внутритекстовый стиль, 44

Г

Граница, 56; 78; 92
  сокращенная форма записи, 29

З

Заголовок, 77
  в закругленной рамке, 81
  определение стиля, 81
  сайта, 173
  уровни, 76
Заполнение, 31; 55; 65; 78; 100
  сокращенная форма, 31
Значение, 24

И

Изображение, 97; 103; 110

К

Каскад, 18
Комментарий, 26

Н

Набор правил, 21

О

Область содержимого, 57
Объявление, 23; 24
Ошибки, 241

П

Панель навигации
  горизонтальная, 149
  вертикальная, 139
Подпись, 102
  рядом с рисунком, 104; 115
Поля, 31; 54
  сокращенная форма, 31
Правило, 21
Псевдокласс, 87; 89

Р

Рамка, 159

С

Свойства, 23
  сокращенная форма записи, 27
Селектор
  идентификаторов, 37
  класса, 36
  объединение, 26
  потомки, 38
  расширенные, 40
  создание, 60
  типа, 36
  универсальные, 39
Сокращенная форма записи, 27; 33
Список, 139; 149
Ссылка, 87
  увеличение активной области, 94
  удаление подчеркивания, 92
Стилевое оформление
  заголовка, 75
Стиль
  внутритекстовый, 44
  заголовков, 45
  применение, 43
  сокрытие от старых браузеров, 47

Т

Таблица, 127
Таблица стилей
  внешняя, 46
Текст, 78
  форматирование, 67

Ф

Фон, 100
  цвет, 55
Фоновое изображение, 55–60; 80; 83; 100
  масштабируемое, 82
Фотогалерея, 107
Фотоснимок, 105

Ш

Шрифт, 76
  общие семейства, 70

Э

Элемент
  блочный, 51
  внутритекстовый, 51

Научно-популярное издание

Расс Уикли

Освой самостоятельно CSS
10 минут на урок

Литературный редактор	*Л.Н. Красножон*
Верстка	*О.В. Мишутина*
Художественный редактор	*В.Г. Павлютин*
Корректор	*А.В. Луценко*

Издательский дом “Вильямс”
101509, Москва, ул. Лесная, д. 43, стр. 1

Подписано в печать 20.03.2006. Формат 84×108/32.
Гарнитура Times. Печать офсетная.
Усл. печ. л. 13,52. Уч.-изд. л. 6,9.
Тираж 3000 экз. Заказ № 809.

Отпечатано с диапозитивов
в ФГУП “Печатный двор” им. А. М. Горького
Федерального агентства по печати и массовым коммуникациям.
197110, Санкт-Петербург, Чкаловский пр., 15.

Шпаргалка CSS

Синтаксис CSS

селектор { свойство: значение; }

Свойства шрифтов

Свойство	Значение	Пример
`font-family`	Название или тип шрифта	`{font-family: arial;}`
`font-style`	`normal | italic`	`{font-style: italic;}`
`font-variant`	`normal | small-caps`	`{font-variant: small-caps;}`
`font-weight`	`normal | bold`	`{font-weight: bold;}`
`font-size`	*процент*|*размер*	`{font-size: 12pt;}`
	`xx-small | x-small`	
	`small | medium | large`	
	`x-large | xx-large`	
	`smaller | larger`	

Свойства текста

Свойство	Значение	Пример
`letter-spacing`	`normal`|*длина*	`{letter-spacing: 5pt;}`
`line-height`	`normal`|*число*	`{line-height: 15pt;}`
	длина|*процент*	
`text-align`	`left | right`	`{text-align: center;}`
	`center | justify`	
`text-decoration`	`none | underline`	
	`overline | line-through`	`{text-decoration:underline;}`
`text-indent`	*длина*|*процент*	`{text-indent: 25px;}`
`text-transform`	`capitalize | uppercase`	`{text-transform: lowercase;}`
	`lowercase | none`	
`vertical-align`	`sub | super`	`{vertical-align: sub;}`
`word-spacing`	`normal` |*длина*	`{word-spacing: 10px;}`

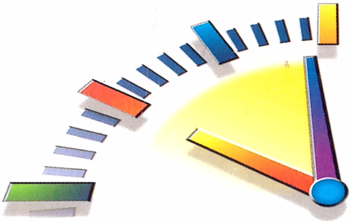
Свойства цвета и фона

Свойство	Значение	Пример
`color`	*Цвет*	`{color: blue;}`
`background-color`	*цвет*|`transparent`	`{background-color: yellow;}`
`background-image`	*url*|`none`	`{background-image: url(marble.gif);}`
`background-repeat`	`repeat | repeat-x repeat-y | no-repeat`	`{background-repeat: no-repeat;}`
`background-attachment`	`scroll | fixed`	`{background-attachment: fixed;}`
`background-position`	`top left | top center top right | center left center center | center right bottom left | bottom center bottom right x% y% | x y`	`{background-position: top left;}`

Шпаргалка CSS

Свойства рамки: границы, поля, заполнение

Свойство	Значение	Пример
border	*ширина*|*стиль*|*цвет*	{border: thick dotted blue;}
border-left	*ширина*|*стиль*|*цвет*	{border-left: thick dotted blue;}
border-right	*ширина*|*стиль*|*цвет*	{border-right: thick dotted blue;}
border-top	*ширина*|*стиль*|*цвет*	{border-top: thick dotted blue;}
border-bottom	*ширина*|*стиль*|*цвет*	{border-bottom: thick dotted blue;}
border-width	thin | medium thick| *длина*	{border-width: thin;}
border-left-width	thin | medium thick| *длина*	{border-left-width: thick;}
border-right-width	thin | medium thick| *длина*	{border-right-width: 0.1in;}
border-top-width	thin | medium thick | *длина*	{border-top-width: 0.5px;}
border-bottom-width	thin | medium thick| *длина*	{border-bottom-width: 0.5cm;}
border-color	*цвет*	{border-color: blue;}
border-left-color	*цвет*	{border-left-color: #0000FF;}
border-right-color	*цвет*	{border-right-color: black;}
border-top-color	*цвет*	{border-top-color: rgb(0,0,255);}
border-bottom-color	*цвет*	{border-bottom-color: red;}
margin-top	*длина*|*процент*|auto	{margin-top: 10px;}
margin-bottom	*длина*|*процент*|auto	{margin-bottom: 1em;}
margin-left	*длина*|*процент*|auto	{margin-left: 5pt;}
margin-right	*длина*|*процент*|auto	{margin-right: 5pt;}
padding-top	*длина*|*длина*	{padding-top: 10px;}
padding-bottom	*длина*|*процент*	{padding-bottom: 1em;}
padding-left	*длина*|*процент*	{padding-left: 5pt;}
padding-right	*длина*|*процент*	{padding-right: 5pt;}



CSS

10 минут на урок

Потратив всего 10 минут на каждый урок, вы научитесь:

- использовать CSS на Web-страницах
- применять стили
- добавлять фоновые изображения
- форматировать текст
- определять стили заголовков и ссылок
- располагать на странице изображения и подписи
- создавать фотогалереи
- форматировать блоки цитат и таблицы
- создавать вертикальную и горизонтальную панели навигации
- настраивать заголовки сайта
- создавать масштабируемую рамку с закругленными углами
- располагать на странице колонки и примечания
- отлаживать код CSS и бороться с ошибками

В этой книге приведены простые практические решения для тех, кто желает быстро получить результаты. В ходе 10-минутных уроков объясняются ключевые концепции CSS и их использование для контроля над визуальным представлением Web-страниц. Кроме того, автор подскажет вам, как избежать пяти печально известных ошибок CSS и сделать ваш код работоспособным в любой ситуации.

Весь представленный в книге код CSS был тщательно протестирован в основных современных браузерах.

Примеры и прочий код, использованный в данной книге, можно загрузить с Web-сайта издательства по адресу: www.williamspublishing.com

Категория: Web-дизайн

Предмет рассмотрения: каскадные таблицы стилей (CSS)

Уровень: для пользователей начального и среднего уровня

ISBN 5-8459-1019-6

06105

9 785845 910196

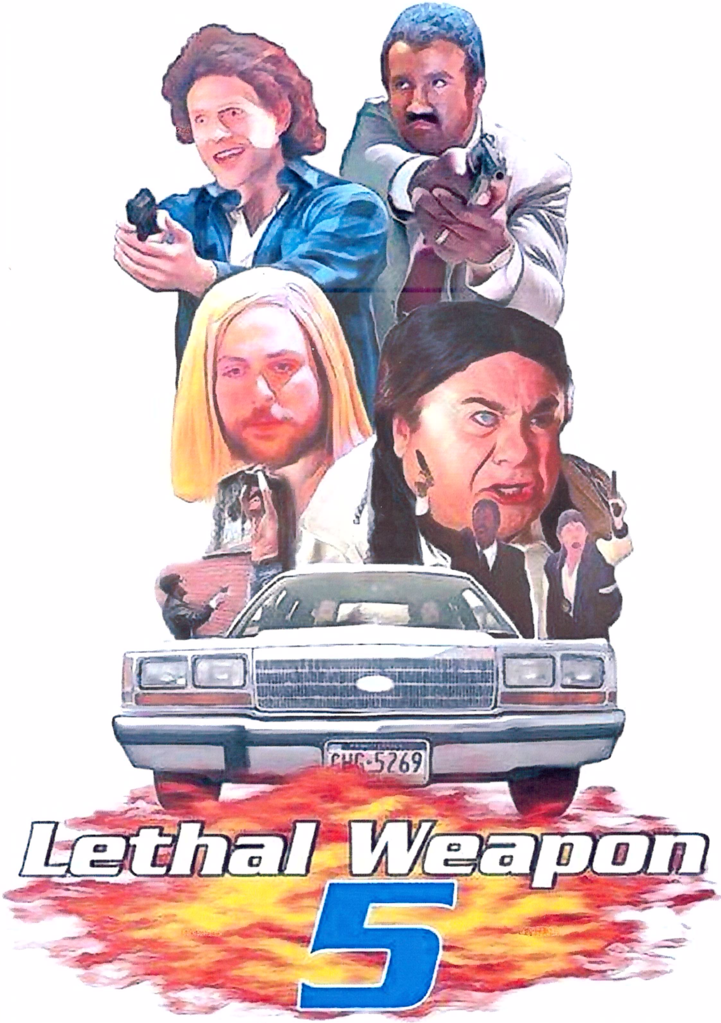
ВИЛЬЯМС

Короткие уроки — быстрые результаты

SAMS
Освой самостоятельно

www.williamspublishing.com
www.samspublishing.com

ВИЛЬЯМС
SAMS
Освой самостоятельно
CSS
Р.Уикли
10 минут на урок



Lethal Weapon 5

IT'S ALWAYS SUNNY IN PHILADELPHIA Presents MAC DENNIS REYNOLDS FRANK REYNOLDS CHARLIE KELLY
Music Composed and Arranged By THE GANG A FRANK REYNOLDS Production
Directed by MAC DENNIS REYNOLDS

R

IT'S ALWAYS SUNNY